DISSERTATION

QUANTIFICATION AND APPLICATION OF UNCERTAINTY IN THE FORMATION OF NANOPARTICLES

Submitted by

Danny Long

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2023

Doctoral Committee:

Advisor: Wolfgang Bangerth

Patrick Shipman
Jiangguo (James) Liu
Richard Finke

Copyright by Danny Long 2023

All Rights Reserved

ABSTRACT

QUANTIFICATION AND APPLICATION OF UNCERTAINTY IN THE FORMATION OF NANOPARTICLES

Nanoparticles are essential across many scientific applications, but their properties are size-dependent. Despite the usefulness of producing monodisperse particle size distributions, it still remains a challenge to fully understand – and hence be able to control – nanoparticle formation reactions due to limitations in what can be observed experimentally. This thesis transfers mathematical, statistical, and computational techniques to this area of nanoparticle chemistry to substantially bolster the sophistication of the quantitative analysis used to better understand nanoparticle systems. First, more efficient software is developed to simulate the reactions. Then, parameter estimation is performed in a robust manner through Bayesian inference, where I demonstrate the ability to parameterize nonlinear ordinary differential equations in such a way that I can fit the observed data and quantify the uncertainty in the parameter estimates. From Bayesian inference, I build three additional analysis frameworks. (1) Model selection through a Bayesian framework; (2) optimizing the yield of the nanoparticle-forming reactions while accounting for uncertainty; and (3) optimizing future measurements to collect data providing the most new information. The culmination of this thesis provides a quantitative framework to analyze arbitrary nanoparticle systems to complement and fill in the gaps of the current experimental techniques.

ACKNOWLEDGEMENTS

Throughout my graduate studies, many people helped me during these stressful few years. First, I want to thank Wolfgang Bangerth, my advisor. You allowed (made) me to choose my own project which allowed my mathematical interests to evolve naturally as well as keep me motivated. Furthermore, you are more than just an academic advisor; you have been a mentor for both my career and personal life, and I am grateful for the growth I have achieved with your help. Thank you as well to the members of Wolfgang's research group for talking about math together and proofreading my papers: Marc, Justin, Tyler, Jerett, Leah, Paddy, Sophie, and Zhuoran.

Thank you to Rick Finke for your presence during my graduate work. I am motivated by applications – rather than purely because math is cool – and your insatiable excitement for science helped tremendously during periods where results were few and far between. Thank you to the rest of the collaborative research group, as well, for the countless conversations about science and providing a diversity of viewpoints: Derek, Chris, Patrick, Luke, and Leah.

The most important thank you is to all of my friends. Life continues during graduate school – except, perhaps, the last couple months leading up to one's defense – and without your support and friendship, graduate school would have been much more challenging. I am grateful to and thankful for you: Brittany, Dustin, Elliot, Erin, Sydney, Elena, Shannon, Lander, Dean, Elle, Lucas, Peter, Steven, and many others.

DEDICATION

I dedicate this thesis to my discovery of coffee during the pandemic.

TABLE OF CONTENTS

ABSTRACT ii
ACKNOWLEDGEMENTS iii
DEDICATION iv
ACRONYMS vii

Chapter 1 Introduction 1

Chapter 2 Mathematical Modeling 4
2.1 The problem: A discussion of the chemical background 4
2.2 The mathematical forward model 7
2.3 Performance considerations 14

Chapter 3 Statistical modeling 17
3.1 Data 19
3.2 Likelihood function 21
3.3 Prior distribution 25
3.4 Exploring the posterior distribution 29
3.5 Statistical inversion of parameters 34
3.5.1 Inversion of the 3-step mechanism 37
3.5.2 Inversion of the 4-step mechanism 42
3.5.3 Data fits 47
3.6 Chain convergence 48
3.7 Model selection 67
3.7.1 Bayes' factor 68
3.7.2 Mixture model 70

Chapter 4 Optimization under uncertainty 76
4.1 Mathematical formulation 77
4.2 Evaluation of the cost function 79
4.3 Optimization algorithm 83
4.3.1 Surrogate model 86
4.3.2 Acquisition function 95
4.4 Optimization of experimental conditions 97
4.4.1 Optimization parameters 98
4.4.2 Optimization visualization 99
4.4.3 Optimization results 103

Chapter 5 Optimal experimental design 110
5.1 Defining optimal experimental design 111
5.2 Computing the utility of an experimental design 115
5.3 Optimal experimental design results 119

Chapter 6 Summary 126

Bibliography 129

ACRONYMS

CDF cumulative distribution function

GP Gaussian process

HME harmonic mean estimator

KDE kernel density estimation

LFIRE Likelihood-Free Inference by Ratio Estimation

MCMC Markov Chain Monte Carlo

MPSRF multivariate potential scale reduction factor

ODEs ordinary differential equations

PDF probability distribution function

PSD particle size distribution

TEM transmission electron microscopy

Chapter 1

Introduction

My dissertation focuses on employing mathematical techniques that aim to quantify the inherent uncertainty associated with real-world problems. Namely, I use techniques from mathematics and Bayesian statistics to answer questions related to Bayesian inversion, optimization under uncertainty, and optimal experimental design, all while incorporating a mathematical model describing chemical reactions that produce nanoparticles. This chapter serves as an introduction to the scientific application and a high-level explanation of the key themes of my work.

Nanoparticles are clusters of matter whose dimensions are in the range of $1\,\mathrm{nm}$-$100\,\mathrm{nm}$ [111]. These particles can be formed out of individual atoms or more complex molecular structures. Nanoparticles are highly useful in industrial applications such as light-emitting diodes [20, 33, 101, 107], solar cells [83, 85, 95, 104], and as catalysts for various chemical reactions [3, 5, 21, 26]. The size of the nanoparticles has a substantial effect in nearly all of these applications. It is not difficult to imagine, then, that being able to modify experimental reaction conditions in order to control the size of the nanoparticles formed is interesting to the community. The first step of this control process is understanding what physical processes are important to the formation of a particular nanoparticle system. If the chemical reactions can be captured mathematically and accurate predictions of the particle size distribution (PSD) are possible, then optimization problems can be formulated to exert control over the system that is desired. The ultimate goal is for a reaction to produce *monodisperse* nanoparticles; that is, particles all of the exact same size. This is not realistic, however, so the practical goal is to produce *near-monodisperse* nanoparticles: particles whose dimensions are $\pm 15\%$ of each other [1].

Mathematical modeling is crucial to this process because the creation of nanoparticles is a highly complex reaction where the number of particles of each possible size influences the outcome of the reaction. Experimental techniques cannot deduce every reaction rate that is present

and a heuristic approach to controlling the shape of the PSD would involve numerous ad hoc manipulations of experimental conditions, which is both time and resource inefficient.

On the other hand, efficient mathematical software can consider all of desired conditions in a fraction of the time. The mathematical modeling technique I utilize is based on *chemical mechanisms*. These mechanisms are simplifications of the chemical reactions that occur. This simplification is accomplished by applying empirical approximations of the relationship between reaction rate for particles of various sizes. As I will discuss later, there are layers of complexity within these mechanisms, with some mechanisms being "more complex" than others. My colleagues and I follow the principle of Occam's razor, meaning we search for the model that is as simple as necessary. The terminology I use in the nanoparticle context is the *minimal mechanism*. Hence, I seek to identify the most simple chemical mechanism that accurately represents the chemical reactions.

The mathematical techniques I employ in my work falls within the realm of Bayesian modeling. These techniques are coined Bayesian from the heavy use of Bayes' theorem [4]. Philosophically, Bayesian modeling considers a data-informed perspective of the probability of an event combined with knowledge of probability of an event known a priori. In settings with discrete possibilities, the combination of this knowledge provides a measure of the probability of an event occurring; with possibilities spanning a continuum, the combined knowledge provides a probability distribution. I eventually want to deduce *parameter values* involved in the mathematical model of the nanoparticle-forming chemical reactions. Instead of identifying point-estimates of these parameter values, Bayesian modeling constructs a probability distribution over the parameter space. Access to a probability distribution allows for the accurate propagation of parameter uncertainty, which allows for more robust simulations of the chemical reactions. As the data for nanoparticle systems are noisy, Bayesian modeling is an effective means to account for this uncertainty introduced by the data.

In Chapter 2 I begin by describing the chemical reactions that form nanoparticles in further detail. From there, I present the mathematical model my colleagues developed and then discuss computational aspects that I implemented to improve the computational performance of the model.

Next, Chapter 3 describes the *Bayesian inverse problem* where I quantify the uncertainty in what parameter values best correspond to experimentally measured data. The result is a set of probability distributions where statistical measures such as mean values and confidence intervals may be extracted. The probability distributions I create can be thought of as a "fitting" procedure. Typically one thinks of an inverse problem as finding model parameters such that the simulation "fits" well with the observed data. In the Bayesian setting, one extends this to testing if an interval of simulation results "fits" well with the observed data. These probability distributions form the basis for the following chapters of my thesis, but I can also use them to select a model of the nanoparticle reactions in a quantitative manner. This model selection is the culmination of the Chapter 3 contents.

Then, Chapter 4 describes an optimization problem to modify the setup of the chemical reaction in order to target a high concentration of a specific particle size while minimizing the spread of the PSD. Moreover, the uncertainty quantification performed in Chapter 3 allows me to frame my work as optimization *under uncertainty* to appropriately account for the uncertainty associated with the model and data.

Finally, the accuracy of simulations depends on how well one can characterize model parameters. Chapter 5 describes the *optimal experimental design* problem which determines how new data should be collected in order to better identify the parameters in the mathematical models; in other words, produce narrower probability distributions during the Bayesian inversion problem of Chapter 3. Narrower probability distributions for parameters will result in narrower confidence intervals in simulations and thus will help reduce uncertainty within questions such as the one I pose in Chapter 4.

Chapter 2

Mathematical Modeling

I begin my technical discussion by developing the mathematical model that I use throughout my thesis. This model was developed in [46] and applied in [47–49, 118]. Although my contributions are not in the development of this model, it is important to understand the model as background for the contributions I do make. To that end, I begin by giving an overview of the relevant chemistry in Section 2.1. Next, in Section 2.2 I describe how to move from the chemistry description of nanoparticles to the mathematical description, ultimately arriving at the mathematical model I use to describe the nanoparticle formation process. Finally, in Section 2.3 I touch on some computational aspects of the mathematical model that were not investigated previously and show how I made substantial improvements in computational performance.

2.1 The problem: A discussion of the chemical background

This section describes the basic chemistry of nanoparticle formation in order to motivate the later chapters of this dissertation. Specifically, the physical processes by which nanoparticles are created and increase in size are presented. This section will not give a comprehensive chemical analysis of nanoparticle science as this dissertation is focused on the mathematical techniques of analyzing and predicting the outcome of these chemical reactions.

Fundamentally, nanoparticles form in three ways:

1. Some precursor compound undergoes a reaction to form the smallest possible nanoparticle – a process called *nucleation*;

2. A precursor reacts with an existing particle, thus increasing the size of the particle – a process called *growth*;

3. Two particles, possibly of different size, react with each other, thus creating a single, larger particle – a process called *agglomeration*.

All of these processes happen at different rates, so if there are nanoparticles of size up to N atoms, then there are N reaction rates describing growth and $N^2/2$ reaction rates describing agglomeration. It is not possible to quantify all of these parameters in any meaningful way. Thus, *pseudo-elementary* steps [7, 8, 27, 28, 55, 62, 116] are proposed to capture the important components of particle formation without introducing over-fitting issues. Pseudo-elementary steps fundamentally apply a growth or agglomeration process to a group of particle sizes in the same way. For example, a system might be characterized by having "small" growth for which particles deemed "small" undergo growth with the same reaction rate. Pseudo-elementary steps, therefore, greatly reduce the number of parameters needed to characterize a set of reactions.

I will use a particular nanoparticle system to demonstrate the techniques I develop. I use a nanoparticle system that has been well-studied [48, 81, 116, and references therein] so that I know what reactions take place, but the contents of this work can be generalized to arbitrary nanoparticle systems so long as the reactions that take place are known. The system I study sees $\{(1{,}5{-}\mathrm{COD})\mathrm{Ir}^{\mathrm{I}}\cdot\mathrm{POM}\}^{8-}$ reduced under $\mathrm{H_2}$, where POM is **p**oly**o**xo**m**etalate, $\mathrm{P_2W_{15}Nb_3O_{62}}^{9-}$, and $1{,}5{-}\mathrm{COD}$ is 1,5-**c**yclo**o**cta**d**iene, $\mathrm{C_8H_{12}}$. This chemical system produces nanoparticles composed of iridium atoms. The nucleation mechanism for this system has been experimentally determined [81] to be

$$\begin{aligned} 2\,[(\mathrm{COD})\,\mathrm{Ir}\,\cdot\mathrm{POM}]^{8-} + 4\,\mathrm{solv} &\underset{k_b}{\overset{k_f}{\rightleftharpoons}} 2\,[(\mathrm{COD})\,\mathrm{Ir(solv)_2}]^{+} + 2\,\mathrm{POM}^{9-}, \\ 2\,[(\mathrm{COD})\,\mathrm{Ir(solv)_2}]^{+} + [(\mathrm{COD})\mathrm{Ir}\,\cdot\mathrm{POM}]^{8-} + 7.5\,\mathrm{H_2} &\xrightarrow{k_1} \mathrm{Ir(0)}_{\,3} + \mathrm{POM}^{9-} + 3\,\mathrm{H}^{+} \\ &\quad + 3\,\mathrm{cyclooctane} + 4\,\mathrm{solv}, \end{aligned} \tag{2.1}$$

where k_* are reaction rates. For brevity, I denote the precursor $[(\mathrm{COD})\,\mathrm{Ir}\,\cdot\mathrm{POM}]^{8-}$ by "A", the solvated complex, $[(\mathrm{COD})\mathrm{Ir(solv)_2}]^{+}$, as "$\mathrm{A(solv)_2}$", the ligand, POM^{9-}, as "L", and (small) particles consisting of iridium(0) atoms generally as "B". The nucleation mechanism (2.1) can then

be equivalently written as

$$\begin{aligned} 2\{\mathrm{A}\ \cdot \mathrm{L} + 2\,\mathrm{solv} &\underset{k_b}{\overset{k_f}{\rightleftharpoons}} \mathrm{A(solv)_2} + \mathrm{L}\}, \\ 2\,\mathrm{A(solv)_2} + \mathrm{A}\ \cdot \mathrm{L} &\xrightarrow{k_1} \mathrm{B_3} + \mathrm{L} + 4\,\mathrm{solv}. \end{aligned} \tag{2.2}$$

For this Ir-POM system, it has been shown in [46, 48, 49] that particle formation can be characterized by "small" and "large" growth, and possibly by "small-small" agglomeration. That is, if we denote "small" particles by "B" and "large" particles by "C", then there are the growth steps

$$\begin{aligned} \mathrm{A} + \mathrm{B} &\xrightarrow{k_2} \mathrm{C} + \mathrm{L}, \\ \mathrm{A} + \mathrm{C} &\xrightarrow{k_3} 1.5\,\mathrm{C} + \mathrm{L}, \end{aligned} \tag{2.3}$$

and potentially the agglomeration step

$$\mathrm{B} + \mathrm{B} \xrightarrow{k_4} \mathrm{C}. \tag{2.4}$$

Therefore, there are two potential mechanistic models to describe this iridium particle system: the *3-step* mechanism

$$\begin{aligned} 2\{\mathrm{A}\ \cdot \mathrm{L} + 2\,\mathrm{solv} &\underset{k_b}{\overset{k_f}{\rightleftharpoons}} \mathrm{A(solv)_2} + \mathrm{L}\}, \\ 2\,\mathrm{A(solv)_2} + \mathrm{A}\ \cdot \mathrm{L} &\xrightarrow{k_1} \mathrm{B_3} + \mathrm{L} + 4\,\mathrm{solv}, \\ \mathrm{A} + \mathrm{B} &\xrightarrow{k_2} \mathrm{C} + \mathrm{L}, \\ \mathrm{A} + \mathrm{C} &\xrightarrow{k_3} 1.5\,\mathrm{C} + \mathrm{L}, \end{aligned} \tag{2.5}$$

and the *4-step* mechanism

$$
\begin{aligned}
2\{\mathrm{A}\cdot\mathrm{L} + 2\,\mathrm{solv} &\underset{k_b}{\overset{k_f}{\rightleftharpoons}} \mathrm{A(solv)_2} + \mathrm{L}\},\\
2\,\mathrm{A(solv)_2} + \mathrm{A}\cdot\mathrm{L} &\xrightarrow{k_1} \mathrm{B_3} + \mathrm{L} + 4\,\mathrm{solv},\\
\mathrm{A} + \mathrm{B} &\xrightarrow{k_2} \mathrm{C} + \mathrm{L},\\
\mathrm{A} + \mathrm{C} &\xrightarrow{k_3} 1.5\,\mathrm{C} + \mathrm{L},\\
\mathrm{B} + \mathrm{B} &\xrightarrow{k_4} \mathrm{C}.
\end{aligned}
\tag{2.6}
$$

Other pseudo-elementary steps have been proposed, but each has been systematically disproven in [46].

I have two chemical models to analyze, but I require a mathematical formulation in order to proceed with the techniques I employ in the following chapters. The next section develops this mathematical formulation.

2.2 The mathematical forward model

In order to analyze (2.5) and (2.6), I need a mathematical model representing the dynamics of these chemical reactions. Modeling chemical reactions follows the *law of mass action* which states that the rate of a chemical reaction is proportional to the product of the concentrations of the reactants. Hence, I utilize a system of ordinary differential equations (ODEs) to represent (2.5) and (2.6). The remainder of this section follows the work in [46]. In the following discussion and the remainder of this work I will omit units from my discussion. However, Table 2.1 provides a reference for the units of all quantities used throughout Chapter 2 – Chapter 5

The law of mass action means if one is modeling the chemical reaction

$$
\alpha\mathrm{X} + \beta\mathrm{Y} \xrightarrow{k} \alpha'\mathrm{X}' + \beta'\mathrm{Y}' \tag{2.7}
$$

Table 2.1: *Reference for the units of parameters I introduce in Chapter 2–Chapter 5.*

Parameter	Description	Units
k_f	Forward reaction rate in (2.2)	$\mathrm{L^2 mol^{-2} h^{-1}}$
k_b	Backward reaction rate in (2.2)	$\mathrm{L mol^{-1} h^{-1}}$
k_1	Nucleation reaction rate in (2.2)	$\mathrm{L^2 mol^{-2} h^{-1}}$
k_2	Small particle growth reaction rate	$\mathrm{L mol^{-1} h^{-1}}$
k_3	Large particle growth reaction rate	$\mathrm{L mol^{-1} h^{-1}}$
k_4	Small particle agglomeration reaction rate	$\mathrm{L mol^{-1} h^{-1}}$
t_*	Time	h
A_0	Initial precursor concentration	$\mathrm{mol L^{-1}}$
POM_0	Initial POM concentration	$\mathrm{mol L^{-1}}$
α_*	Multiplicative factor to parameter $*$	unitless

where X, Y, X′, and Y′ are chemical species and $\alpha, \beta, \alpha', \beta'$ are integers, then the resulting equations describing the time evolution of the chemical concentrations are

$$\begin{aligned}
\frac{dX}{dt} &= -k\alpha X^{\alpha} Y^{\beta}, \\
\frac{dY}{dt} &= -k\beta X^{\alpha} Y^{\beta}, \\
\frac{dX'}{dt} &= k\alpha' X^{\alpha} Y^{\beta}, \\
\frac{dY'}{dt} &= k\beta' X^{\alpha} Y^{\beta}.
\end{aligned} \tag{2.8}$$

If (2.7) was the reversible reaction

$$\alpha \mathrm{X} + \beta \mathrm{Y} \underset{k_b}{\overset{k_f}{\rightleftharpoons}} \alpha' \mathrm{X}' + \beta' \mathrm{Y}' \tag{2.9}$$

then instead of (2.8), the time evolution of the chemical species is described with

$$\begin{aligned}
\frac{dX}{dt} &= -k_f \alpha X^{\alpha} Y^{\beta} + k_b \alpha X'^{\alpha'} Y'^{\beta'}, \\
\frac{dY}{dt} &= -k_f \beta X^{\alpha} Y^{\beta} + k_b \beta X'^{\alpha'} Y'^{\beta'}, \\
\frac{dX'}{dt} &= k_f \alpha' X^{\alpha} Y^{\beta} - k_b \alpha' X'^{\alpha'} Y^{\beta'}, \\
\frac{dY'}{dt} &= k_f \beta' X^{\alpha} Y^{\beta} - k_b \beta' X'^{\alpha'} Y^{\beta'}.
\end{aligned} \tag{2.10}$$

Finally, the superposition principle applies to a series of chemical reactions. That is, the rates of change from each reaction can be added together to quantify the rates of change for the entire reaction. For example, the set of reactions

$$\begin{aligned} \mathrm{X} + \mathrm{Y} &\xrightarrow{k_1} \mathrm{Z}, \\ \mathrm{X} + \mathrm{Z} &\xrightarrow{k_2} \mathrm{Y} \end{aligned} \tag{2.11}$$

is modeled by

$$\begin{aligned} \frac{dX}{dt} &= -k_1 XY - k_2 XZ, \\ \frac{dY}{dt} &= -k_1 XY + k_2 XZ, \\ \frac{dZ}{dt} &= k_1 XY - k_2 XZ, \end{aligned} \tag{2.12}$$

where the first term of each right-hand side comes from the first reaction of (2.11) and the second term in each right-hand side is from the second reaction.

These modeling principles from the law of mass action can be applied to the mechanisms (2.5) and (2.6) [46, 48, 49]. The following mathematical notation is used: the precursor species $\mathrm{A} \cdot \mathrm{L}$ as A, the solvent solv as S, the solvated complex $\mathrm{A(solv)_2}$ as A_s, the particles represented by B and C as B_i where i is the size of the particle, and the ligand L as L.

The nucleation mechanism is given by

$$\begin{aligned} 2\{\mathrm{A} \cdot \mathrm{L} + 2\,\mathrm{solv} &\underset{k_b}{\overset{k_f}{\rightleftharpoons}} \mathrm{A(solv)_2} + \mathrm{L}\}, \\ 2\,\mathrm{A(solv)_2} + \mathrm{A} \cdot \mathrm{L} &\xrightarrow{k_1} \mathrm{B}_3 + \mathrm{L} + 4\,\mathrm{solv}. \end{aligned}$$

Note that the concentration of the solvent is not tracked over time. This is because the amount of solvent is large compared to the other chemical species and, experimentally, the concentration of the solvent does not significantly change over time. Hence, S is treated as a constant. The

pseudo-elementary “small” growth mechanism

$$\mathrm{A} + \mathrm{B} \xrightarrow{k_2} \mathrm{C} + \mathrm{L}$$

is described by the individual reactions

$$\mathrm{A} + \mathrm{B}_i \xrightarrow{\alpha_i k_2} \mathrm{B}_{i+1} + \mathrm{L}, \qquad 3 \leq i \leq M$$

where M is the cutoff between “small” and “large” particles, B_i represents a particle of size i and

$$\alpha_i = i r_i$$

is a function describing how many atoms are on the surface of the particle, thus allowing the interaction of this chemical reaction to take place anywhere on the surface. The function

$$r_i = 2.677 i^{-0.28}$$

is an empirical description of what proportion of atoms are located on the surface of a nanoparticle [96]. Similarly, the “large” growth mechanism

$$\mathrm{A} + \mathrm{C} \xrightarrow{k_3} 1.5\,\mathrm{C} + \mathrm{L}$$

represents the reactions

$$\mathrm{A} + \mathrm{B}_i \xrightarrow{\alpha_i k_3} \mathrm{B}_{i+1} + \mathrm{L}, \qquad M < i \leq J,$$

where J is the largest particle size. This notation is made more concise by defining

$$\beta_i = \begin{cases} k_2\alpha_i, & 3 \leq i \leq M, \\ k_3\alpha_i, & M < i \leq J \end{cases}$$

and considering the reactions

$$\mathrm{A} + \mathrm{B}_i \xrightarrow{\beta_i} \mathrm{B}_{i+1} + \mathrm{L}, \qquad 3 \leq i \leq J.$$

The agglomeration mechanism

$$\mathrm{B} + \mathrm{B} \xrightarrow{k_4} \mathrm{C}$$

represents the reactions

$$\mathrm{B}_i + \mathrm{B}_j \xrightarrow{\alpha_i\alpha_j k_4} \mathrm{B}_{i+j}, \qquad 3 \leq i, j \leq M.$$

Again, I make this more concise by defining

$$\Gamma_{ij} = \begin{cases} k_4\alpha_i\alpha_j, & 3 \leq i \leq M, \\ 0, & M < i \leq J \end{cases}$$

and using the reactions

$$\mathrm{B}_i + \mathrm{B}_j \xrightarrow{\Gamma_{ij}} \mathrm{B}_{i+j}, \qquad 3 \leq i, j \leq J.$$

Tying everything together, the equations for the 3-step mechanism (2.5) are

$$
\begin{aligned}
\frac{dA}{dt} &= \underbrace{-k_f A S^2 + k_b A_s L - k_1 A A_s^2}_{\text{nucleation}} - \underbrace{A \sum_{j=3}^{J} \beta_j B_j}_{\text{growth}}, \\
\frac{dA_s}{dt} &= \underbrace{k_f A S^2 - k_b A_s L - 2k_1 A A_s^2}_{\text{nucleation}}, \\
\frac{dL}{dt} &= \underbrace{k_f A S^2 - k_b A_s L}_{\text{nucleation}} + \underbrace{A \sum_{j=3}^{J} \beta_j B_j}_{\text{growth}}, \\
\frac{dB_3}{dt} &= \underbrace{k_1 A A_s^2}_{\text{nucleation}} - \underbrace{\beta_3 A B_3}_{\text{growth}}, \\
\frac{dB_i}{dt} &= \underbrace{A(\beta_{i-1} B_{i-1} - \beta_i B_i)}_{\text{growth}}, \qquad 4 \le i \le J,
\end{aligned}
\tag{2.13}
$$

and the equations for the 4-step mechanism (2.6) are

$$
\begin{aligned}
\frac{dA}{dt} &= \underbrace{-k_f A S^2 + k_b A_s L - k_1 A A_s^2}_{\text{nucleation}} - \underbrace{A \sum_{j=3}^{J} \beta_j B_j}_{\text{growth}}, \\
\frac{dA_s}{dt} &= \underbrace{k_f A S^2 - k_b A_s L - 2k_1 A A_s^2}_{\text{nucleation}}, \\
\frac{dL}{dt} &= \underbrace{k_f A S^2 - k_b A_s L}_{\text{nucleation}} + \underbrace{A \sum_{j=3}^{J} \beta_j B_j}_{\text{growth}}, \\
\frac{dB_3}{dt} &= \underbrace{k_1 A A_s^2}_{\text{nucleation}} - \underbrace{\beta_3 A B_3}_{\text{growth}} - \underbrace{\sum_{j=3}^{M} \Gamma_{3j} B_3 B_j}_{\text{agglomeration}}, \\
\frac{dB_i}{dt} &= \underbrace{A(\beta_{i-1} B_{i-1} - \beta_i B_i)}_{\text{growth}} \\
&\quad \underbrace{- \sum_{j=3}^{M} \Gamma_{ij} B_i B_j + \sum_{\substack{\mu+\nu=i \\ 3 \le \mu \le \nu \le M}} \Gamma_{\mu\nu} B_m u B_n u}_{\text{agglomeration}}, \qquad 4 \le i \le J.
\end{aligned}
\tag{2.14}
$$

Equations (2.13) and (2.14) define the dynamics of the chemical reactions and, with appropriate choice of parameters k_*, M, it is possible to predict the PSD if the appropriate mechanism is identified, as is demonstrated in Figure 2.1. Further details can be found in [46, 48, 49].

In (2.13) and (2.14) the parameter J defines the largest particle that can be created or, rather, the largest particle that is meaningful to track. The size of the system of ODEs – and therefore the computational expense of performing simulations such as those in Figure 2.1 – is dependent on J. For the iridium system introduced in Section 2.1, $J = 2500$. This selection of $J = 2500$ comes from the observation that experiments usually involve particles up to approximately $4\,\mathrm{nm}$ and a particle composed of 2500 iridium atoms corresponds to approximately $4\,\mathrm{nm}$ (see (3.4) for this conversion).

Now that the equations governing nanoparticle formation have been derived in (2.13) and (2.14), I want to discuss computational performance and my first contribution to this work.

2.3 Performance considerations

In this section, I discuss the performance of solving the ODEs in (2.13) and (2.14). The initial work done on this project by my colleagues did not require high levels of performance to have practical feasibility. However, the work I will perform in upcoming chapters will require multiple orders of magnitude more solves of the ODEs than has been done so far. Thus, my first contribution is to translate a few practices from the computational sciences to speed up the solves of the ODEs I have to compute.

In previous work [46–49, 118], the implementation to solve (2.13) and (2.14) was performed in MATLAB using the ODE15S function [97]. This solver is chosen because the ODEs are stiff – a general term indicating the necessity of excessively small step sizes required for stability when using "normal" ODE solvers – and therefore specialized methods are necessary for efficient solves. Most notably, stiff solvers such as ODE15S use *implicit* methods which require the solution of a linear system involving the Jacobian matrix – the partial derivatives of each right-hand side function in the ODEs – at every time step. By default in MATLAB, this linear system is formed by taking numerical derivatives if the Jacobian matrix is not provided; this is the case in this initial implementation. Taking numerical derivatives rather than using analytically derived equations provides robustness to errors since hand-computing the Jacobian is an error-prone process, but sacrifices computational performance. This implementation, on modern hardware, takes 40–90 seconds to solve the ODEs, with the time variance relating to the values of the parameters k_*, M.

MATLAB provides options to specify the Jacobian matrix and use sparse matrices to improve performance. However, I decided to utilize a C++ implementation to achieve performance improvements. Comparisons between my C++ implementation versus using ODE15S with a provided Jacobian and sparse matrices might yield comparable performance results, but this comparison is out of the scope of my work.

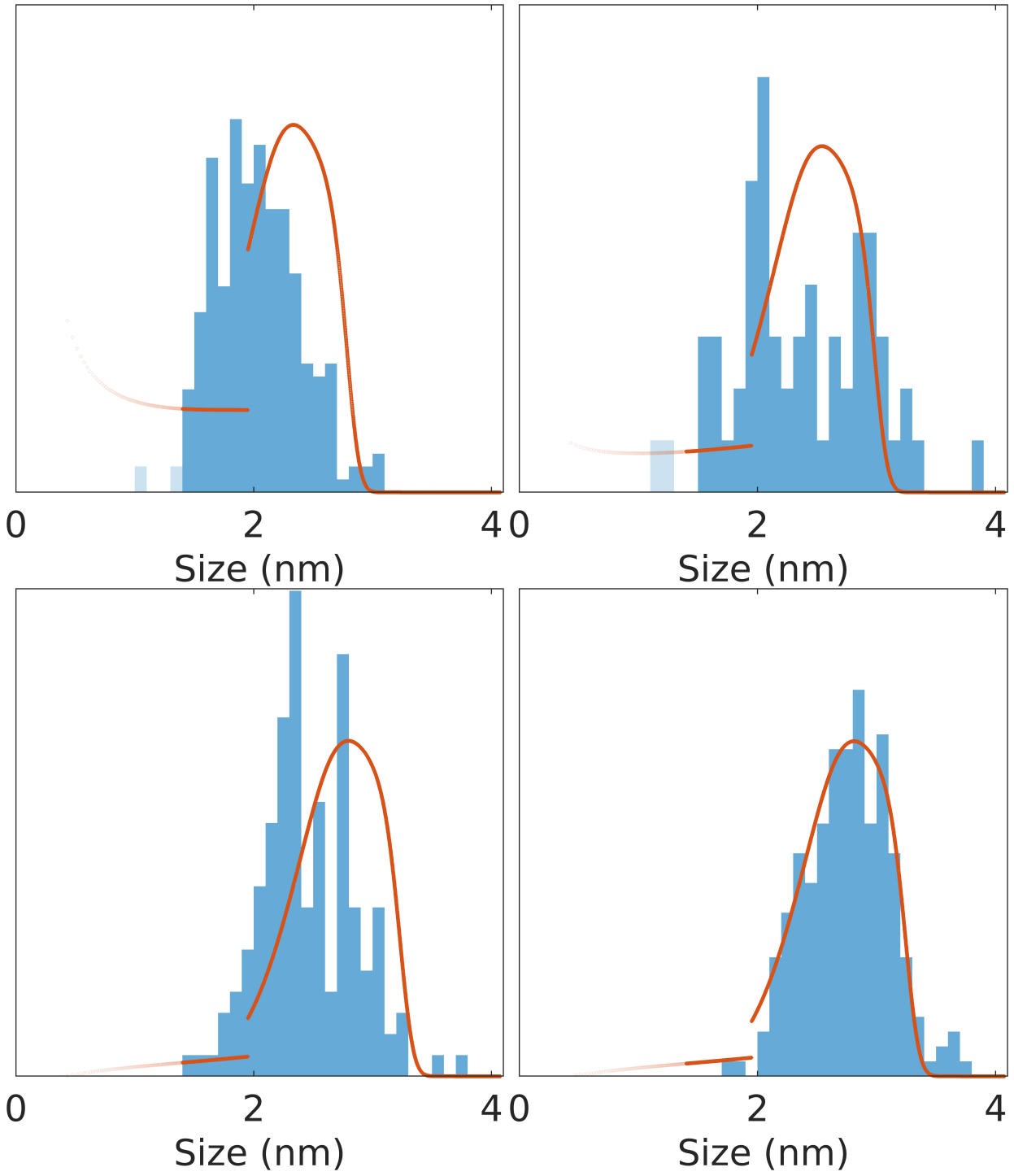


Figure 2.1: *Plots showing the measured PSD of* $\mathrm{Ir}(0)_n$ *with a blue histogram versus the simulated PSD in a red curve. The simulated PSD is produced using the optimized parameters in [49] with the 3-step mechanism* (2.5)*. From left to right and top to bottom, the PSDs are at times 0.918, 1.170, 2.336, and 4.838 hours. The y-axis is a probability density whose precise value is not meaningful in this context, hence labels are omitted.*

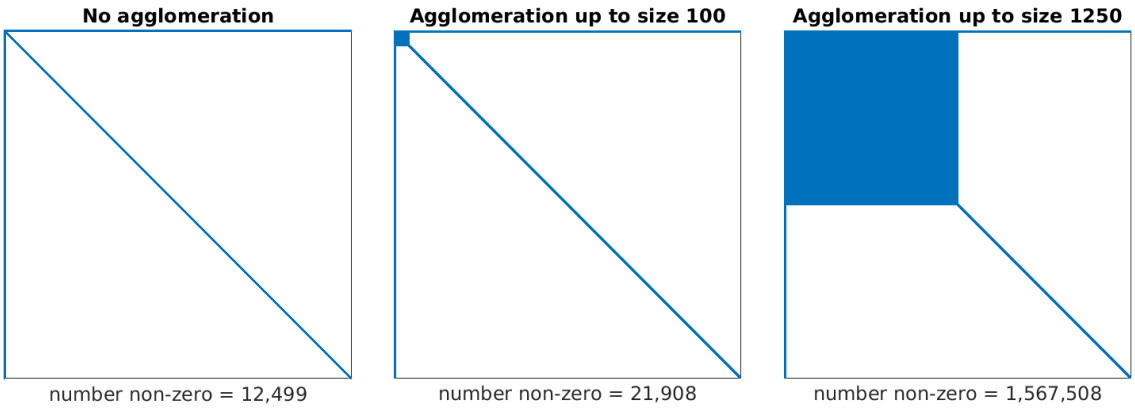


Figure 2.2: *Plots showing the sparsity pattern of the Jacobian matrix. (Left) The sparsity pattern when no agglomeration is present, that is, the 3-step mechanism; (Middle) The sparsity pattern when agglomeration can occur between particles with 100 atoms or fewer; (Right) The sparsity pattern when agglomeration can occur between particles with 1250 atoms or fewer. This rightmost plot shows the most dense possible Jacobian that I model since agglomeration between two 1250 atom particles results in a 2500 atom particle, the largest such particle I consider.*

Using a C++ implementation that uses EIGEN [40] for efficient linear algebra operations and SUNDIALS [52] for state-of-the-art ODE solvers, the system of ODEs in (2.13) or (2.14) can be solved in 1–5 seconds on modern hardware. This timeline is reasonable even for the computationally expensive simulations required in Chapter 3. These performance benefits can be attributed to noting the sparsity in the Jacobian matrix, as seen in Figure 2.2, which led me to use sparse matrices and sparse linear solvers, which are more efficient than their dense linear solver counterparts (assuming a matrix is "sparse enough"). EIGEN provides many ways to solve linear systems, but I found the BiCGSTAB method [110] using an incomplete LU factorization [92] as a preconditioner resulted in the best performance.

Now that I can solve (2.13) and (2.14) in a few seconds, I am ready to fit my models to the observed data. I will perform this fitting procedure in a Bayesian manner, which will require solving the ODEs tens of thousands of times in serial, hence why I needed to improve performance. The contents of Chapter 3 explains this fitting procedure in detail.

Chapter 3

Statistical modeling

This chapter describes the process of performing parameter estimation in a way that gives statistical information about the parameters. This process is called Bayesian inversion and this methodology has seen large uptake in the last twenty years in a broad spectrum of disciplines. Specifically in chemistry, a literature review finds examples of Bayesian inversion in work such as [6, 9, 11, 16, 18, 25, 31, 32, 34, 44, 45, 57, 59, 69, 84, 86, 94, 100, 108, 121]. Furthermore, a two-part review performed by Armstrong and Hibbert [2, 51] details the use of Bayesian inversion in the chemistry sub-disciplines of combustion, chemometrics, forensic sciences, medical testing, microbiology/DNA analysis, chromatography and mass spectrometry, environmental chemistry, and occupational health and safety. That being said, Bayesian inversion techniques have not yet propagated into nanoparticle science and thus the work of my dissertation and the paper published [70] along the way are the first examples of the use of Bayesian inversion in the nanoparticle context, to my knowledge.

The research previously performed by my colleagues in [46–49, 118] exemplifies the traditional – that is, *deterministic* – approach to inverse problems, which are able to produce results such as those in Figure 2.1. This type of inverse problem requires relatively few evaluations of the forward model (solving the system of ODEs from (2.13) or (2.14)), but is only able to produce a point-estimate of the parameters. That is, only the single, best set of parameters is identified with no indication of variability. Conversely, Bayesian inversion requires a large number of forward evaluations, but in-turn provides an entire distribution of the parameter space. Thus, the point-estimate would correspond to the mode of the distribution, but Bayesian inversion also provides information about the uncertainty of each parameter and I can make decisions based on interpreting this uncertainty. Sections 3.5–3.7 go through this decision-making process.

Bayesian inversion answers the question *to what certainty are the parameters known?* This process uses Bayes' theorem which states that

$$\pi(\mathcal{K}|\text{data}) = \frac{\pi_{\text{L}}(\text{data}|\mathcal{K})\pi_{\text{pr}}(\mathcal{K})}{\pi(\text{data})}, \tag{3.1}$$

where:

- $\pi(\mathcal{K}|\text{data})$ is called the *posterior distribution* and quantifies the probability a set of parameters $\mathcal{K}$ is "correct" based on the available data;
- $\pi_{\text{L}}(\text{data}|\mathcal{K})$ is called the likelihood distribution and quantifies how likely it is to have measured the available data if a particular set of parameters is assumed to be true;
- $\pi_{\text{pr}}(\mathcal{K})$ is called the *prior distribution* and quantifies any prior knowledge about what, if any, parameter values I *believe* to be true;
- $\pi(\text{data})$ is a normalization factor.

To highlight that the data is a constant and that I do not care about the normalization factor of the likelihood distribution, I will instead use the *likelihood function* $L(\mathcal{K}) \propto \pi_{\text{L}}(\text{data}|\mathcal{K})$ for the remainder of this chapter. As I will demonstrate in Section 3.4, I only actually care about the value of the posterior distribution up to a constant – that is, I only care about the *ratio* of posterior values when developing probability distributions – and thus I will consider the proportionality

$$\pi(\mathcal{K}|\text{data}) \propto L(\mathcal{K})\pi_{\text{pr}}(\mathcal{K}). \tag{3.2}$$

The goal of Bayesian inversion is to estimate $\pi(\mathcal{K}|\text{data})$. In general, there is not an analytic expression for the likelihood function and, consequently, the posterior distribution. Thus, numerical algorithms are required to construct the posterior and to perform the subsequent data analysis. The remainder of this chapter accomplishes these tasks by:

1. Analyzing the available experimental data in Section 3.1 to motivate my derivations of the likelihood and prior;

2. Developing a likelihood function in Section 3.2 that represents the data collection process and the associated measurement error;

3. Constructing a prior distribution in Section 3.3 that encodes my beliefs about what parameter values are reasonable;

4. Describing a class of numerical algorithms in Section 3.4 that approximate the posterior distribution;

5. Perform the Bayesian inversion in Section 3.5 to create probability distributions for $\mathcal{K}$;

6. Analyze the posterior samples in Section 3.6 to assess the convergence of the numerical algorithm;

7. Compare the posterior samples from the 3-step and 4-step mechanisms in Section 3.7 to quantitatively select the more appropriate model.

I begin by discussing the experimental data I have.

3.1 Data

Bayesian inversion relies on understanding how likely it is to have measured a certain set of data if a set of parameters is assumed to reflect physical reality. Therefore it is crucial to understand the data collection process in order to construct a likelihood function that provides insight about the posterior distribution. This section discusses the data for the iridium nanoparticle system I am analyzing.

The key quantity of interest for nanoparticle systems is the PSD. The primary way to collect data about the PSD is a measurement technique called transmission electron microscopy (TEM). Essentially, a beam of electrons is transmitted through the nanoparticle solution and an image is produced. The image contains pictures of the nanoparticles and the diameters of these particles

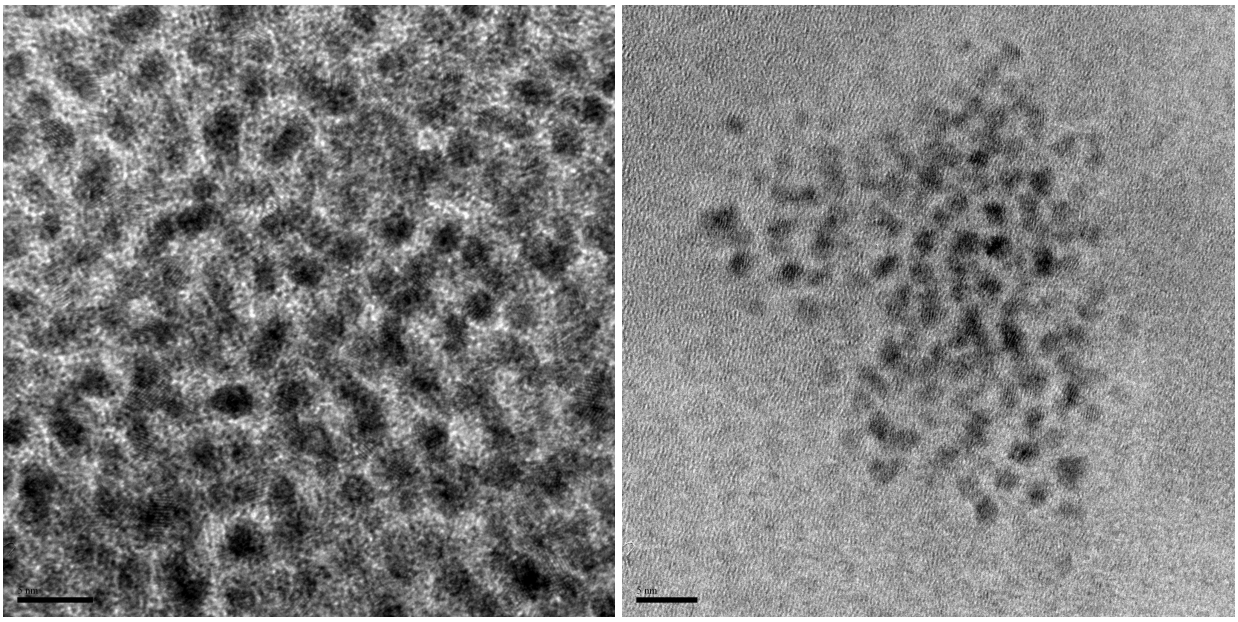


Figure 3.1: *Examples of the collected TEM data. Images are provided from [117, supplemental information].*

can be measured with the aid of computer software. Examples of these images are in Figure 3.1. Measurements are conducted at multiple different times throughout the chemical reaction in order to get an idea of the time change of the PSD. Time series data is crucial to finding appropriate chemical reaction rates because, in the absence of multiple time points of data, there is no distinction between the reaction finishing at its actual finish time and the reaction finishing at any time before the true reaction time. Hence, in this scenario without time series data, if parameters $\mathcal{K}^*$ are optimal, then $\lambda\mathcal{K}^*$ for $\lambda \geq 1$ are also optimal.

There are two ways to get multiple measurements: (i) the chemical reaction is started and subsequently stopped when the measurement is taken, then additional chemical reactions are started from scratch with the same experimental conditions to gather additional data; or (ii) small samples are removed from the solution throughout the reaction and measurements are taken on those samples. The data used in this dissertation used methodology (ii). Either way the data is collected, it is reasonable to assume that each measurement is independent from each other because the number of extracted particles is substantially less than the total number of particles and the reacting solution is well-mixed. With an assumption of independence between data sets, I can consider a joint

likelihood function for all of the data sets by simply multiplying the likelihood functions for single data sets.

The collected data is summarized into the histograms in Figure 3.2. The histograms are presented such that detected particles smaller than $1.4\,\mathrm{nm}$ are displayed in light blue while larger particles are shown in dark blue. This distinction indicates the limitations of the measurement process for smaller sized particles. Therefore, I only consider comparisons between the data and simulations for particle sizes larger than $1.4\,\mathrm{nm}$.

The presentation of the data in Figure 3.2 provides the key to how to construct the likelihood function. If I simulate a PSD and convert it to a probability distribution binned into the same histogram rectangles as the data, then I can compare the expected frequency of each bin of particle sizes in the simulation to the observed frequency of each bin of particle sizes. I now formalize this idea into a mathematical expression.

3.2 Likelihood function

With the type of data collected understood, I derive an appropriate likelihood function. The purpose of the likelihood function is to convert the solution of the forward model into a form that is comparable to the data. At its core, the likelihood function assumes $\mathcal{K}$ are the true values of the parameters and it produces a quantitative measure of the (unnormalized) probability of measuring the observed data under this assumption. Assuming a $\mathcal{K}$ in this context requires solving equations (2.13) or (2.14) in order to determine the particle concentrations B_i. Since I assume a set of parameters, I need to make a probability distribution based off the result of this simulation. This can be readily constructed by creating the probability mass function

$$p_i = B_i \left(\sum_{i=3}^{J} B_i \right)^{-1}, \qquad 3 \leq i \leq J. \tag{3.3}$$

The probability distributions in (3.3) are specified for each particle size. However, the data presented in Section 3.1 are binned together based on the particle diameter into a histogram. The

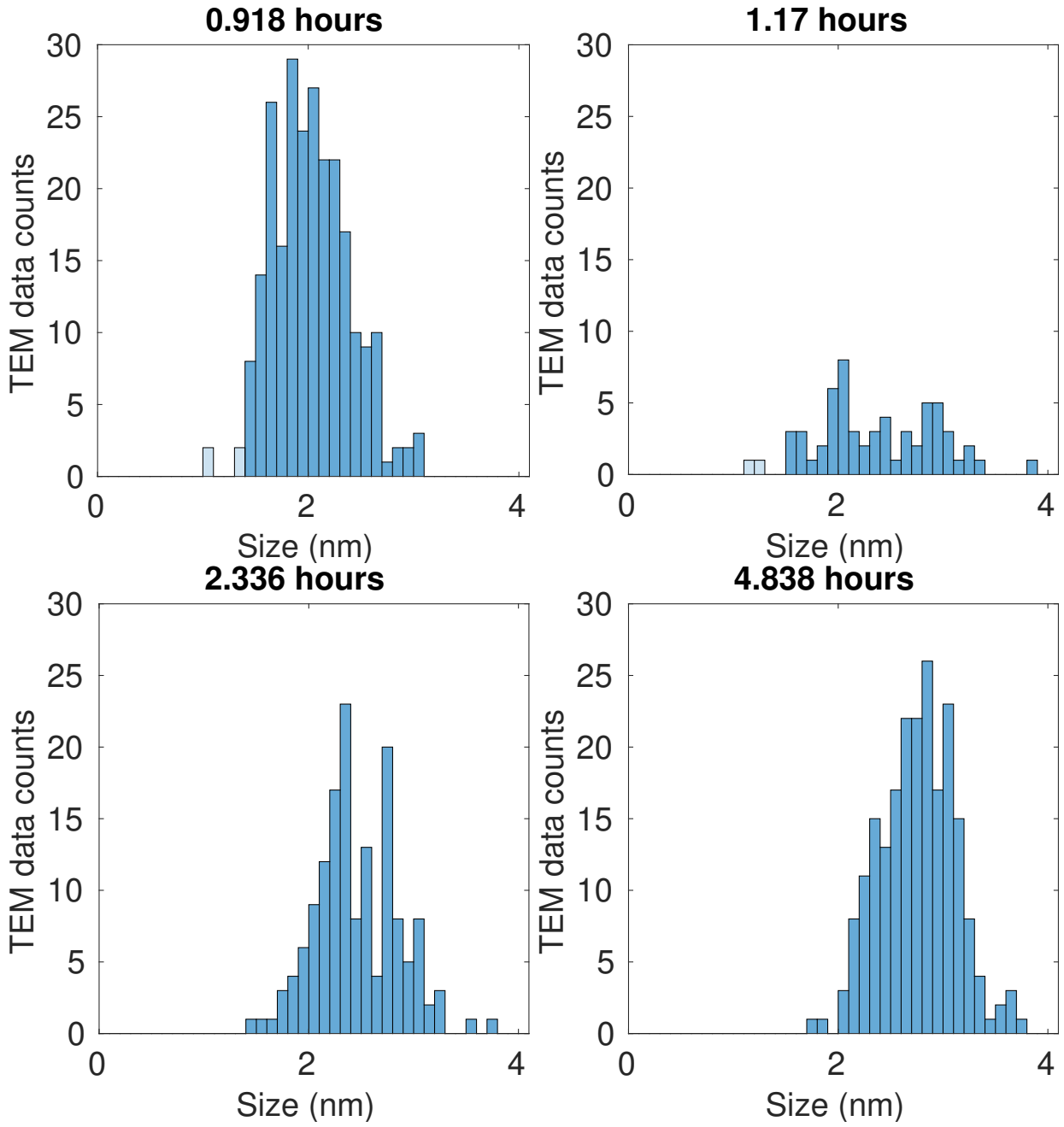


Figure 3.2: *Histograms showing the number of* $\mathrm{Ir}(0)_n$ *particles measured via transmission electron microscopy at different time points of the reaction (based on [117]). Particle counts for particles smaller than* $1.4\,\mathrm{nm}$ *are unreliable and are shown in a lighter color. For comparison, nanoparticles composed of* $J = 2500$ *iridium atoms have a size of* $4.0\,\mathrm{nm}$.

histogram bins start at $1.4\,\mathrm{nm}$ (see Section 3.1 for an explanation of this lower bound), end at $4.1\,\mathrm{nm}$, and each have width $0.1\,\mathrm{nm}$. Therefore, I need to approximate the diameter corresponding to each particle size and then bin the probabilities into a probability mass function corresponding to the same histogram bins the data uses.

The diameter can be estimated by the equation

$$\text{diameter}(i) = 0.3000805 \times i^{1/3}, \tag{3.4}$$

where i is the number of atoms in a particle. This empirical equation is based on a nonlinear fit of published size data for $\text{Ir}(0)_i$ particles, where i is the number of iridium atoms, which can be found in [48, Figure S1]. Using (3.4), I can congregate (3.3) into the binned probability mass function by defining

$$p_{\beta_\ell} = \sum_{\text{diameter}(i)\in\beta_\ell} p_i, \tag{3.5}$$

where β_ℓ are the bin ranges defined by $\beta_1 = [1.4, 1.5), \beta_2 = [1.5, 1.6), \cdots, \beta_{26} = [3.9, 4.0), \beta_{27} = [4.0, 4.1]$. These binned probabilities correspond to measured data in each bin denoted b_ℓ^{measured}.

Identifying a single particle from the TEM data is essentially a random selection of a particle labeled with some size bin β_ℓ. Hence, this measurement process is akin to picking colored balls out of an urn, which is the simile I will use to ease the interpretation of my derivation of the likelihood function.

Suppose the balls can be any of n different colors and I randomly select a set of b balls that happens to have b_i balls of color i, where $i = 1, \cdots, n$. Let N be the total number of balls in the urn, such that $N \gg b = b_1 + \cdots + b_n$. Let p_i be the probability of a ball having color i in the urn. The process begins by selecting b of the N balls, of which each selection has probability N^{-1}. Now that a selection of balls is chosen, b_1 of the b balls are selected to be color 1. Each of these selections occur independently with probability p_1; since $N \gg b$, assuming this process is independent and that p_1 stays constant is reasonable and holds for the subsequent steps as well.

Then, from the remaining $b - b_1$ balls, select b_2 of the balls to be color 2. These balls are each selected with probability p_2. If I continue this process until all colors are accounted for, I find the probability of selecting balls $\{b_i\}$ given probabilities $\{p_i\}$ and that b total balls are selected to be

$$p\left(\{b_i\}|\{p_i\}, b \text{ selected}\right) = \frac{1}{N^b}\binom{N}{b}\binom{b}{b_1}p_1^{b_1}\binom{b-b_1}{b_2}p_2^{b_2}\cdots\binom{b-b_1-\cdots-b_{n-1}}{b_n}p_k^{b_n}, \quad (3.6)$$

where $\binom{n}{k}$ represents the binomial coefficient. I can simplify (3.6) by reducing the binomial coefficients to

$$\begin{aligned}\binom{N}{b}\binom{b}{b_1}\cdots\binom{b-b_1-\cdots-b_{n-1}}{b_n} &= \frac{N!}{(N-b)!b!}\frac{b!}{(b-b_1)!b_1!}\cdots\frac{(b-b_1-\cdots-b_{n-1})!}{(b-b_1-\cdots-b_n)!b_n!} \\ &= \frac{N!}{(N-b)!}\prod_{i=1}^{n}\frac{1}{b_i!}.\end{aligned} \quad (3.7)$$

Thus I can represent the probability (3.6) as

$$p\left(\{b_i\}|\{p_i\}, b \text{ selected}\right) = \left(\frac{1}{N^b}\frac{N!}{(N-b)!}\prod_{i=1}^{n}\frac{1}{b_i!}\right)\prod_{i=1}^{n}\pi_i^{b_i} \quad (3.8)$$

Now, using the notation developed for the nanoparticle system, I make the substitutions $p_\ell = p_{\beta_\ell}(\mathcal{K})$ to use the probabilities in (3.5) and to note the parameter dependence of this quantity, $b_\ell = b_\ell^{\text{measured}}$ to use the number of measured particles in each bin, $N = N_{\text{total}}$ to represent to total number of particles in the chemical solution, and let b to represent the total number of particles collected in the measurement. Hence, I have

$$p(\text{data}|\mathcal{K}) = \underbrace{\left(\frac{1}{N_{\text{total}}^b}\frac{N_{\text{total}}!}{(N_{\text{total}}-b)!}\prod_{\ell=1}^{27}\frac{1}{b_\ell^{\text{measured}}!}\right)}_{\text{independent of }\mathcal{K}}\underbrace{\prod_{\ell=1}^{27}p_{\beta_\ell}(\mathcal{K})^{b_\ell^{\text{measured}}}}_{\mathcal{K}\text{ dependent}}. \quad (3.9)$$

For the likelihood function I want to use to approximate the posterior distribution, normalization factors are irrelevant. Here, normalization factors are any quantity independent of the variable parameters $\mathcal{K}$. Hence I can remove the $\mathcal{K}$-independent term noted in (3.9) to get my likelihood

function

$$L(\mathcal{K}) = \prod_{\ell=1}^{27} p_{\beta_\ell}(\mathcal{K})^{b_\ell^{\text{measured}}}. \tag{3.10}$$

For practical purposes, (3.10) is not ideal. In general, $p_{\beta_\ell}(\mathcal{K}) \ll 1$ and $b_\ell^{\text{measured}} \gg 1$ so the likelihood calculation is prone to rounding error when implemented in software. Hence what I actually use in my software is

$$\log L(\mathcal{K}) = \sum_{\ell=1}^{27} b_\ell^{\text{measured}} \log p_{\beta_\ell}(\mathcal{K}). \tag{3.11}$$

The likelihood function I derived is only for a single data set. As I discussed in Section 3.1, I can assume independence of the data sets and therefore independence of the likelihood functions. If $L_i(\mathcal{K})$ is the likelihood function for the ith set of data, then I can use independence – and the fact that I have four data sets for my problem – to get

$$\begin{aligned} L(\mathcal{K}) &= \prod_{i=1}^{4} L_i(\mathcal{K}), \\ \log L(\mathcal{K}) &= \sum_{i=1}^{4} \log L_i(\mathcal{K}). \end{aligned} \tag{3.12}$$

I now have derived the likelihood function. Therefore I have one of the ingredients to (3.2). The other ingredient to compute the posterior distribution is the prior distribution.

3.3 Prior distribution

Now that the likelihood function has been derived, the next step is to develop the prior distribution. Pairing the prior distribution and the likelihood function together allows me to use the algorithms, which I describe in Section 3.4, to estimate the posterior distribution. This section discusses my choice of the prior distribution.

The prior distribution captures the "expert" knowledge about the parameters in question. That is, if there are certain values of the parameters that do not make physical sense, then that can be encoded by assigning them a probability of zero or a very low probability. On the other hand, if experimental results or other scientific intuition suggest certain parameter values are more likely, then these parameter values can be given a higher probability in order to put more trust in them. That being said, when I say the prior distribution is "expert" knowledge, that is to be interpreted as a very high standard. If there is not a large degree of confidence in specifying a prior distribution, then it is better to simply use an uninformative prior distribution. An example of how a prior distribution that does not accurately reflect the posterior performs worse than an uninformative prior distribution can be seen in [103, Example 4.66].

In this context, an uninformative prior distribution means a uniform distribution. Thus all parameter values within the range of the uniform distribution are deemed "equally likely". Since neither I nor my chemist colleagues have reason to suspect certain ranges of parameter values are more likely than others, I take the uninformative prior distribution route in order to not influence the posterior distribution. For particular parameters $\mathcal{K}\{k_1, k_2, \cdots, k_N\}$ where N is the number of parameters, the uniform distribution for the ith parameter of $\mathcal{K}$ is defined by its probability density function

$$\text{Unif}_{k_i}^{\text{continuous}}(a,b) = \begin{cases} \frac{1}{b-a}, & a \leq k_i \leq b, \\ 0, & \text{otherwise} \end{cases} \tag{3.13}$$

for continuous parameters and its probability mass function

$$\text{Unif}_{k_i}^{\text{discrete}}(a,b) = \begin{cases} \frac{1}{b-a+1}, & a \leq k_i \leq b, \qquad k_i \in \mathbb{Z}, \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

for discrete (integer) parameters.

The question, then, is what should be the bounds for the uniform distribution for each parameter. My approach is to pick the bounds such that the prior distribution does not impede the exploration of the posterior distribution. Since the posterior distribution is proportional to the product of the likelihood function and the prior distribution, any region of parameter space where the prior distribution is zero results in the posterior distribution also being zero. For the nanoparticle system described in Chapter 1, the parameters space for each mechanism is

$$\begin{aligned} \mathcal{K}_{\text{3-step}} &= \{k_f, k_b, k_1, k_2, k_3, M\}, \\ \mathcal{K}_{\text{4-step}} &= \{k_f, k_b, k_1, k_2, k_3, k_4, M\}. \end{aligned} \tag{3.15}$$

Each of the k_* parameters are reaction rates that take on values in $\mathbb{R}$. M is the cutoff between "small" and "large" particles as described in Chapter 1; thus M takes integer values. For the rate constants, negative values do not make physical sense. For the particle size cutoff, integer values less than the smallest particle size do not make sense – for the iridium system, the smallest particle size is three. Now I simply need upper bounds on the uniform distributions that are large enough to not affect results. I performed some preliminary (partial) runs of the algorithms described in Section 3.4 with various values for the upper bound until I found the order of magnitude for each parameter value that was never reached in my simulations. Then I added an additional order of magnitude to be conservative. For the M parameter, my chemist colleagues have chemical reasons for why the cutoff should be at most a particle with 600 atoms, although simply choosing the largest particle size (2500 atoms in this case) is also a fine option; in my simulations, for which results can be found in Section 3.5, a particle with 600 atoms was reached with negligible frequency and thus I am confident that the choice of 600 instead of 2500 did not affect my results. As a result, the prior distributions I use are given in (3.16). For ease of notation Unif_M is taken to be the discrete uniform probability mass function in (3.14) and Unif_{k_*} are taken to be the continuous uniform

probability density function in (3.13). I then use the following:

$$\begin{aligned}
\pi_{\text{pr}}^{\text{3-step}}(\mathcal{K}) &= \text{Unif}_{k_f}(0, 1\times10^3)\times\text{Unif}_{k_b}(0, 2\times10^8)\times\text{Unif}_{k_1}(0, 1\times10^8)\\
&\quad\times\text{Unif}_{k_2}(0, 1\times10^8)\times\text{Unif}_{k_3}(0, 1\times10^8)\times\text{Unif}_{M}(3, 600),\\
\pi_{\text{pr}}^{\text{4-step}}(\mathcal{K}) &= \text{Unif}_{k_f}(0, 1\times10^3)\times\text{Unif}_{k_b}(0, 2\times10^8)\times\text{Unif}_{k_1}(0, 1\times10^8)\\
&\quad\times\text{Unif}_{k_2}(0, 1\times10^8)\times\text{Unif}_{k_3}(0, 1\times10^8)\times\text{Unif}_{k_4}(0, 1\times10^8)\\
&\quad\times\text{Unif}_{M}(3, 600).
\end{aligned} \tag{3.16}$$

The use of prior distribution (3.16) describes the approach I take for my first round of analysis, as described in Section 3.5 and visualized in Figure 3.3 and Figure 3.6. At that point in my analysis, although the results of analysis A are not sufficient for characterizing the parameters as I will discuss later, the generated posterior distributions are my best estimation of what the parameter space looks like. The ultimate role of the prior distributions is to encode my beliefs in what the posterior distribution is. Hence, after analysis A, from a Bayesian perspective I have updated my beliefs and now, instead of taking the uninformative prior distribution as my initial beliefs, I take the results of analysis A as what I believe the posterior to be. Therefore, I perform subsequent rounds of analysis where the posterior distribution from the previous analysis informs the new prior distribution. One could form this new prior distributions with a kernel density estimation (KDE) [82, 90], but since all of the evidence I have points to my posterior being unimodal and in order to make the prior computations easier, I use the multivariate normal distribution

$$\pi_{\text{pr}}^{(2)}(\mathcal{K}) = (2\pi)^{-p/2}\left|\Sigma\right|^{-1/2}\exp\left\{-\frac{1}{2}\left(\mathcal{K}-\mu\right)^{\intercal}\Sigma^{-1}\left(\mathcal{K}-\mu\right)\right\}, \tag{3.17}$$

where p is the number of parameters ($p = 6$ for the 3-step model and $p = 7$ for the 4-step model) and μ and Σ are sample mean and covariance matrix of the samples generated in the previous analysis. Technically, the multivariate normal distribution allows negative values (or values < 3 in the case of the parameter M), which are nonphysical in my problem. To remedy this, I can bound

the prior distribution away from the nonphysical parameter values and consider

$$\begin{aligned} NF &= \int_{-\infty}^{0}\int_{-\infty}^{3} \pi_{\text{pr}}^{(2)} dk_* dM + \int_{-\infty}^{0}\int_{2500}^{\infty} \pi_{\text{pr}}^{(2)} dk_* dM \\ \pi_{\text{pr}}(\mathcal{K}) &= \frac{1}{1-NF}\pi_{\text{pr}}^{(2)}(\mathcal{K}). \end{aligned} \tag{3.18}$$

Notice that the parameter M is now treated as a continuous value. This assumption is fine as the value of M can be rounded down for a physical interpretation within the mathematical model (but the unrounded number must be kept within posterior samples!). This point is also discussed in Section 3.4 and the numerical results in Section 3.5 concur that $M \in \mathbb{R}^+$ instead of $M \in \mathbb{Z}^+$ does not deteriorate my results.

Now that the likelihood function and the prior distribution have been selected, I can use established algorithms to create an approximation of the posterior distribution. These algorithms are discussed in the next section.

3.4 Exploring the posterior distribution

I now have both necessary ingredients to (3.2) and therefore I have a way to calculate the posterior distribution, $\pi(\mathcal{K}|\text{data})$, for individual values of $\mathcal{K}$ (up to an unimportant factor). However, in theory, I need to perform this calculation for *every possible parameter value* in order to have the exact posterior distribution. This is not possible since there are infinitely many $\mathcal{K}$. Therefore, I need to utilize algorithms to approximate the posterior distribution based on a finite number of evaluated $\mathcal{K}$.

A popular way to approximate the posterior distribution in this manner is a class of algorithms called Markov Chain Monte Carlo (MCMC) methods. At its core, this class of methods considers parameters $\mathcal{K}^{(i)}$ and, based on only $\mathcal{K}^{(i)}$ (hence Markov Chain), proposes a trial sample $\mathcal{K}^{(i+1)}$ by randomly perturbing $\mathcal{K}^{(i)}$ (hence Monte Carlo). Then the trial sample is accepted or rejected based off some criteria that varies between algorithms. The result is a chain of samples $\left\{\mathcal{K}^{(0)}, \mathcal{K}^{(1)}, \mathcal{K}^{(2)}, \cdots\right\}$. The posterior distribution is then constructed with the N samples gener-

ated by weighting each sample by N^{-1}. Statistical quantities can then be computed with

$$\bar{k}_b \approx \frac{1}{N} \sum_{i=1}^{N} k_b^{(i)} \pi(\mathcal{K}^{(i)}|\text{data})$$

using N samples, and similar expressions for the mean values of other parameters, covariances, and other quantities of interest. The efficacy of these approximations depends on the MCMC algorithm retaining many samples $\mathcal{K}^{(i)}$ where $\pi(\mathcal{K}|\text{data})$ is large and few samples where the posterior distribution is small.

The first MCMC algorithm I employ is called Metropolis-Hastings [50]. This algorithm is as follows:

1. Propose a trial sample $\mathcal{K}^{\text{trial}}$, chosen in some proximity of the current sample $\mathcal{K}^{\text{current}}$ via the *proposal distribution* $g(\mathcal{K}^{\text{current}})$.

2. Evaluate the ratio of probabilities

$$AR = \frac{\pi(\mathcal{K}^{\text{trial}}|\text{data}) g(\mathcal{K}^{\text{current}}|K^{\text{trial}})}{\pi(\mathcal{K}^{\text{current}}|\text{data}) g(\mathcal{K}^{\text{trial}}|K^{\text{current}})} \tag{3.19}$$

   to calculate an acceptance ratio.

3. Accept or reject $\mathcal{K}^{\text{trial}}$ with probability $\min(1, AR)$. In the case of acceptance, $\mathcal{K}^{\text{trial}}$ is appended to the list of collected samples. In the case of rejection, $\mathcal{K}^{\text{current}}$ is appended to the list of collected samples.

4. Repeat for a specified number of samples N.

In the calculation of the acceptance ratio (3.19), the ratio of the posterior distribution favors samples with higher posterior distribution values. On the other hand, the ratio of the proposal distribution works to mitigate the effect of the proposal distribution favoring certain parameter values. In other words, if the proposal distribution is not symmetric, the acceptance ratio takes this into consideration. In practice, the proposal distribution is typically taken to be a uniform or

normal distribution. Both of these distributions are symmetric and thus

$$\begin{aligned} 1 &= \frac{g(\mathcal{K}^{\text{current}}|\mathcal{K}^{\text{trial}})}{g(\mathcal{K}^{\text{trial}}|\mathcal{K}^{\text{current}})}, \\ AR &= \frac{p(\mathcal{K}^{\text{trial}}|\text{data})}{p(\mathcal{K}^{\text{current}}|\text{data})}. \end{aligned} \tag{3.20}$$

I use a multivariate normal proposal distribution, so I am able to use (3.20).

The total acceptance rate across an entire chain of samples is often tracked as a measure of how efficient the sampling procedure is. It can be shown in specific circumstances that an acceptance rate of 23.4% is optimal [36]. Therefore, with the Metropolis-Hastings algorithm, it is common practice to tune the hyperparameters of the proposal distribution to target a 23.4% acceptance rate. While this result is proven for an idealized and largely unrealistic example, practical experience shows that targeting a 23.4% acceptance rate works quite well for a large class of problems [12, Section 4.2]. That being said, practical wisdom also says that an acceptance rate in the range of approximately $15\% - 50\%$ works well enough, so I do not have to be too precise with my hyperparameter tuning. I use an uncorrelated multivariate normal distribution of the form

$$g_{\text{3-step}}(\mathcal{K}^{\text{trial}}|\mathcal{K}^{\text{current}}) \sim N\left(\mathcal{K}^{\text{current}}, \begin{bmatrix} \sigma^2_{k_f} & & & & & & \\ & \sigma^2_{k_b} & & & & & \\ & & \sigma^2_{k_1} & & & & \\ & & & \sigma^2_{k_2} & & & \\ & & & & \sigma^2_{k_3} & & \\ & & & & & \ddots & \\ & & & & & & \sigma^2_M \end{bmatrix}\right), \tag{3.21}$$

$$g_{\text{4-step}}(\mathcal{K}^{\text{trial}}|\mathcal{K}^{\text{current}}) \sim N\left(\mathcal{K}^{\text{current}}, \begin{bmatrix} \sigma^2_{k_f} & & & & & & & \\ & \sigma^2_{k_b} & & & & & & \\ & & \sigma^2_{k_1} & & & & & \\ & & & \sigma^2_{k_2} & & & & \\ & & & & \sigma^2_{k_3} & & & \\ & & & & & \sigma^2_{k_4} & & \\ & & & & & & \sigma^2_{M} \end{bmatrix}\right). \tag{3.22}$$

for the proposal distribution.

There are numerous improvements over the Metropolis-Hastings algorithm that have been developed [39,42,43,53,113]. The purpose of my dissertation is not to do an exhaustive comparison of MCMC methods, but after using Metropolis-Hastings to get an initial posterior distribution as seen in Section 3.5, I want to perform subsequent MCMC iterations for additional analyses. For these MCMC runs, I want to improve my convergence to the posterior distribution to save computational time. Hence I use the adaptive variant of Metropolis-Hastings [43] as well.

Adaptive Metropolis-Hastings proposes a simple modification of Metropolis-Hastings. The key observation is when a new sample is proposed according to the proposal distribution, if samples are proposed from a distribution similar to the posterior, then the sampler is more efficient. In this context, more efficient means fewer samples are required for reasonable convergence to the posterior. In the original Metropolis-Hastings algorithm, one typically has to manually tune various hyperparameters – such as the *fixed* proposal distribution – in order to achieve efficient sampling. Adaptive Metropolis-Hastings, on the other hand, learns the shape of the posterior distribution as samples are generated and adaptively updates the proposal distribution with this accumulated knowledge. Typically the algorithm is of the form:

1. Begin with a user-defined proposal distribution and user-defined "learning period" composed of n samples to learn the posterior shape.

2. After n samples, compute the current sample covariance matrix C_i and propose a trial sample according to $\mathcal{K}^{\text{trial}} \sim N(\mathcal{K}^{\text{current}}, \delta C_i + \varepsilon I)$, where $N(\cdot, \cdot)$ is the multivariate normal distribution, δ is some multiplier to control how close to propose samples, and εI is a small multiple of the identity matrix used for numerical stability.

3. Perform the normal Metropolis-Hastings logic to accept or reject $\mathcal{K}^{\text{trial}}$.

4. Update the sample covariance matrix and repeat.

Theoretical analysis of this algorithm on test problems shows that $\delta = 2.38^2/(\text{no. of parameters})$ is a good choice [37]. That being said, as my results in Section 3.5 show, this rule of thumb is not always optimal because the theoretical analysis required knowledge of the true posterior distribution. Hence, after I perform an analysis using adaptive Metropolis-Hastings, I do a final analysis using traditional Metropolis-Hastings to flesh out my results. This behavior I see is also why often the adaptive Metropolis-Hastings algorithm is often presented as the algorithm I list above plus a final step of fixing the covariance matrix after N iterations and performing the Metropolis-Hastings algorithm afterwards. The only modification I make to this is updating my prior beliefs in between and performing manual hyperparameter tuning to the proposal distribution to get a more optimal acceptance ratio.

Also note that the adaptivity is applied through a continuous random variable, rather than the discrete random variable used for parameter M. One could either model M as a continuous random variable and truncate it to an integer after a sample is proposed or split the proposal distribution into an adaptive proposal for the continuous parameters and a separate – adaptive or not adaptive! – proposal for the discrete parameters. For simplicity, I choose to extend M to be a real number instead of just an integer and round down when applying it in the model. This choice also pairs well with using the multivariate normal distribution as a prior as I describe in Section 3.3.

In this section I described how to estimate the posterior distribution using an iterative method. With this, I have all of the components needed to sample from the posterior distribution. In the next section, I will perform this sampling.

3.5 Statistical inversion of parameters

In the previous sections of this chapter I have explained how to perform a Bayesian inversion analysis. I developed the likelihood function and the prior distribution specific to the nanoparticle problem discussed in Chapter 1. This section brings the entire process together and demonstrates my results of performing Bayesian inversion on the iridium nanoparticle system. I present my results separately for the 3-step mechanism in Section 3.5.1 and the 4-step mechanism in Section 3.5.2. All results in this chapter use the MCMC algorithm implementations in the UQLAB software package [72, 114]. First, however, I will describe the general procedure I conducted for both analyses. I first perform a hyperparameter-tuning procedure wherein I:

1. Run the Metropolis-Hastings algorithm for a few short chains;
2. Document the acceptance ratio of those chains;
3. Broaden the proposal distribution if the acceptance ratios are larger than 23.4%, or narrow the proposal distribution if the acceptance ratios are smaller than 23.4%;
4. Repeat 1–3 until all of the chains achieve an acceptance ratio close to 23.4%, for which I used $\pm 10\%$ as my "sniff test";
5. Begin the full run of 232 chains each containing 20000 samples.

The results using this procedure are documented in [70]. In this paper my colleagues and I published, there are two differences from what I will present in this work:

- I did not perform a quantitative assessment of model fit and thus did not save the data regarding (unnormalized) posterior density values;
- I assumed the ratio between k_f and k_b was known and fixed for simplicity.

That being said, my initial analysis [70] accomplished the non-trivial task of finding an area of parameter space that explains the data quite well. However, I do want evaluations of the posterior density for the model comparisons I will discuss in Section 3.7 and I would like an analysis

where k_f and k_b are allowed to vary independently of each other. As a result, I need to redo my analysis. This time, however, I am not starting from scratch. I have developed knowledge about the parameter space. I have a sense of the shape of the posterior distribution now and I can encode that in the proposal distribution used in the Metropolis-Hastings algorithm. Based on the results I produced [70, eqns. 16, 17], I have that

$$\begin{aligned} K^*_{\text{3-step}} = \big\{ & k_f^* = (3.31 \pm 0.38) \times 10^{-3}, \quad k_b^* = (6.62 \pm 0.75) \times 10^3, \\ & k_1^* = (1.24 \pm 0.12) \times 10^5, \quad k_2^* = (2.60 \pm 0.86) \times 10^5, \\ & k_3^* = (6.22 \pm 0.25) \times 10^3, \quad M^* = 107 \pm 5 \big\}, \end{aligned} \tag{3.23}$$

$$\begin{aligned} K^*_{\text{4-step}} = \big\{ & k_f^* = (6.85 \pm 0.65) \times 10^{-2}, \quad k_b^* = (1.37 \pm 0.13) \times 10^5, \\ & k_1^* = (7.69 \pm 0.86) \times 10^4, \quad k_2^* = (1.40 \pm 0.10) \times 10^4, \\ & k_3^* = (7.15 \pm 0.32) \times 10^3, \quad k_4^* = (1.74 \pm 0.78) \times 10^3, \\ & M^* = 111 \pm 14 \big\}, \end{aligned} \tag{3.24}$$

where the values of k_f come from the relation $k_f = k_b \times 5 \times 10^{-7}$ and the reported error is the sample standard deviation. For both the 3- and 4-step mechanisms, I do the following:

- Construct covariance matrices $C_{\text{3-step}}$ and $C_{\text{4-step}}$ based on the standard deviations in (3.23) and (3.24), excluding the correlation terms;
- Generate starting parameters for 20 chains by drawing from multivariate normal distributions centered about the mean values in (3.23) and (3.24), with covariance of $1.1C_{\text{3-step}}$ and $1.1C_{\text{4-step}}$; the factor of 1.1 is introduced because a convergence criteria described in Section 3.6 is dependent on the starting points for each chain to be dispersed more broadly than the posterior;
- Use the uniform prior distributions in (3.16);
- Set the proposal distribution to be of the form of (3.21) and (3.22) with the covariance matrices satisfying $\alpha C_{\text{3-step}}$ and $\beta C_{\text{4-step}}$;

- Starting with $\alpha, \beta = 1$, run 20 chains of 100 samples and document the acceptance ratio of each chain;
- Adjust α, β until all 20 chains have acceptance ratios of 20%–30%;
- Using the tuned α, β, perform the Metropolis-Hastings algorithm [50] to generate 15000 samples.

I consider this procedure "analysis A" and describe it as such in the figures and tables I present. Following this, I perform the adaptive Metropolis-Hastings procedure with:

- The prior distribution as a multivariate normal with mean and covariance matrix derived from the last 50% of the posterior samples in analysis A – the first 50% of samples are "burned" in order to forget where the chains started;
- 20 chains worth of initial samples generated from the prior distribution, except with the covariance scaled up by 10%;
- The proposal distribution following the covariance matrix of the prior distribution, but scaled based on the same procedure to target the first 300 samples of each chain having acceptance ratios in the 20%–30% range;
- An initial mixing period of 300 samples;
- An adaptation period of 14700 samples.

I consider the result of this procedure "analysis B". Finally, I take this second set of results and repeat "analysis B" except with the non-adaptive Metropolis-Hastings algorithm to form "analysis C". The three analyses I conduct are summarized in Table 3.1. Now, I begin by summarizing the results of these three analyses for the 3-step mechanism.

Table 3.1: *Summary of the MCMC analyses I perform. Adaption period refers to the number of samples with an adaptive covariance matrix.* $N(\cdot,\cdot)$ *refers to the normal distribution.* $\mathbb{E}(\cdot)$ *refers to the sample mean.* $\mathcal{C}(\cdot)$ *refers to the sample covariance matrix.*

	Chains	Samples	Adaptive	Adaptation Period	Prior Distribution	Posterior Samples
Analysis A	20	15000	No	–	Uniform	$\mathcal{K}^{(A)}$
Analysis B	20	15000	Yes	14700	$N\left(\mathbb{E}(\mathcal{K}^{(A)}), \mathcal{C}(\mathcal{K}^{(A)})\right)$	$\mathcal{K}^{(B)}$
Analysis C	20	15000	No	–	$N\left(\mathbb{E}(\mathcal{K}^{(B)}), \mathcal{C}(\mathcal{K}^{(B)})\right)$	$\mathcal{K}^{(C)}$

Table 3.2: *Summary of the central tendencies of the posterior distribution for MCMC analysis A of the 3-step mechanism using the Metropolis-Hastings algorithm. MAP is the maximum a posteriori, the maximum value observed for the posterior distribution. The posterior distribution is visualized in Figure 3.3.*

Parameter	MAP	Mean	$2.5\% - 97.5\%$ Quantile
k_f	1.2×10^{-3}	2.2×10^{-3}	$(1.2 - 4.1) \times 10^{-3}$
k_b	7.3×10^{3}	7.0×10^{3}	$(5.4 - 8.8) \times 10^{3}$
k_1	58.7×10^{4}	26.8×10^{4}	$(8.5 - 57.1) \times 10^{4}$
k_2	16.4×10^{5}	13.0×10^{5}	$(2.7 - 32.9) \times 10^{5}$
k_3	5.6×10^{3}	5.8×10^{3}	$(5.4 - 6.4) \times 10^{3}$
M	109	106	$96 - 115$

3.5.1 Inversion of the 3-step mechanism

Using the 3-step mechanism, I perform analysis A. Figure 3.3 shows the resulting marginal posterior distributions and Table 3.2 summarizes the central tendencies and the 95% credible interval of each parameter. For this analysis, I have three notable observations:

1. Parameter k_1 has a very broad distribution, as identified in the third row of Table 3.2 showing the 95% credible interval spanning two orders of magnitude. Moreover, the *maximum a posteriori* (i.e. the best observed posterior density value) is outside the credible interval. This indicates a lack of convergence to the posterior; more on this in Section 3.6.

2. Parameter k_2 also has a very broad distribution, again which can be seen by the credible interval in the fourth row of Table 3.2 spanning two orders of magnitude.

3. There is an interesting correlation between parameters k_1 and k_f as seen in the marginal distribution in the first column and third row of Figure 3.3.

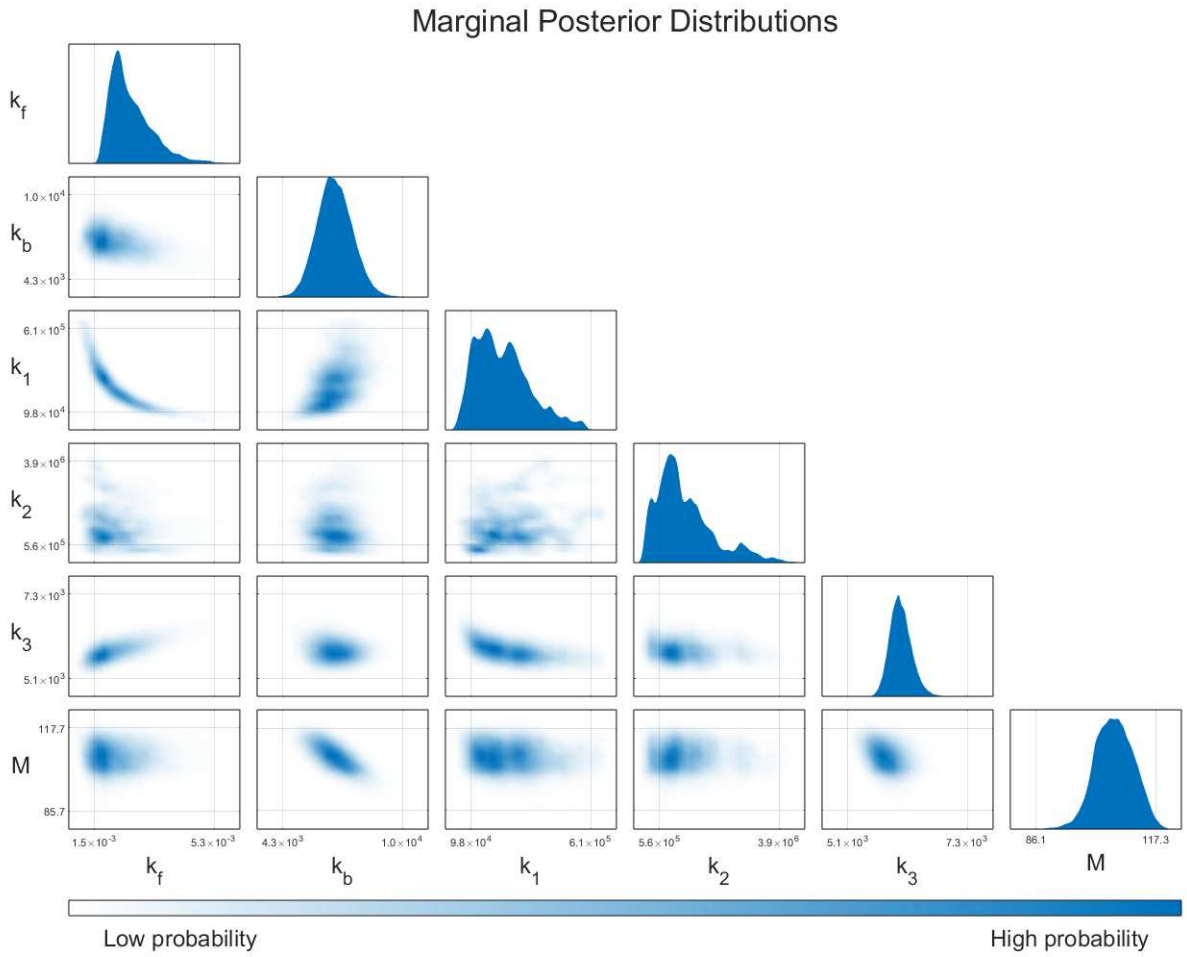


Figure 3.3: *1- and 2-dimensional marginal distributions for MCMC analysis A of the 3-step mechanism using the Metropolis-Hastings algorithm. The diagonal shows the 1-dimensional marginals and the lower-triangular regions shows the 2-dimensional marginal as indicated by the row and column labels. Summary statistics are presented in Table 3.2.*

Table 3.3: *Summary of the central tendencies of the posterior distribution for MCMC analysis B of the 3-step mechanism using the Metropolis-Hastings algorithm. MAP is the maximum a posteriori, the maximum value observed for the posterior distribution. The posterior distribution is visualized in Figure 3.4.*

Parameter	MAP	Mean	$2.5\% - 97.5\%$ Quantile
k_f	1.2×10^{-3}	1.2×10^{-3}	$(1.0 - 1.5) \times 10^{-3}$
k_b	7.5×10^{3}	7.5×10^{3}	$(6.4 - 8.6) \times 10^{3}$
k_1	5.9×10^{5}	5.8×10^{5}	$(4.3 - 7.3) \times 10^{5}$
k_2	18.5×10^{5}	18.4×10^{5}	$(6.0 - 32.3) \times 10^{5}$
k_3	5.6×10^{3}	5.6×10^{3}	$(5.4 - 5.9) \times 10^{3}$
M	109	107	$100 - 114$

Next, analysis B of the 3-step mechanism produces the posterior distribution shown in Figure 3.4 and the summarized parameter information in Table 3.3. Here, my observations are:

1. The distribution for k_2 remains quite broad, as can be seen as its credible interval in Table 3.3 still encompasses two orders of magnitude.

2. The distribution for k_1 has settled close to the *maximum a posteriori* found in analysis A, supporting the idea that analysis A failed to converge to the posterior distribution.

3. The one dimensional marginal distributions seen on the diagonal of Figure 3.4 are rather jagged. This indicates a low acceptance probability; more on this and better visualizations of this phenomenon in Section 3.6.

Finally, I perform analysis C of the 3-step mechanism. The posterior marginal distributions are shown in Figure 3.5 and parameter summaries are in Table 3.4. My observations are:

1. The distribution of k_2 remains broad, spanning two orders of magnitude. As I discuss in Section 3.6, I am confident analysis C converged to the posterior distribution and thus I conclude that k_2 is the most challenging parameter to identify in this model.

2. The jaggedness of the one dimensional marginal distributions have been smoothed out as compared to analysis B. This indicates a more optimal acceptance ratio, although a similar effect could be produced by "thinning" the chains; more on this in Section 3.6.

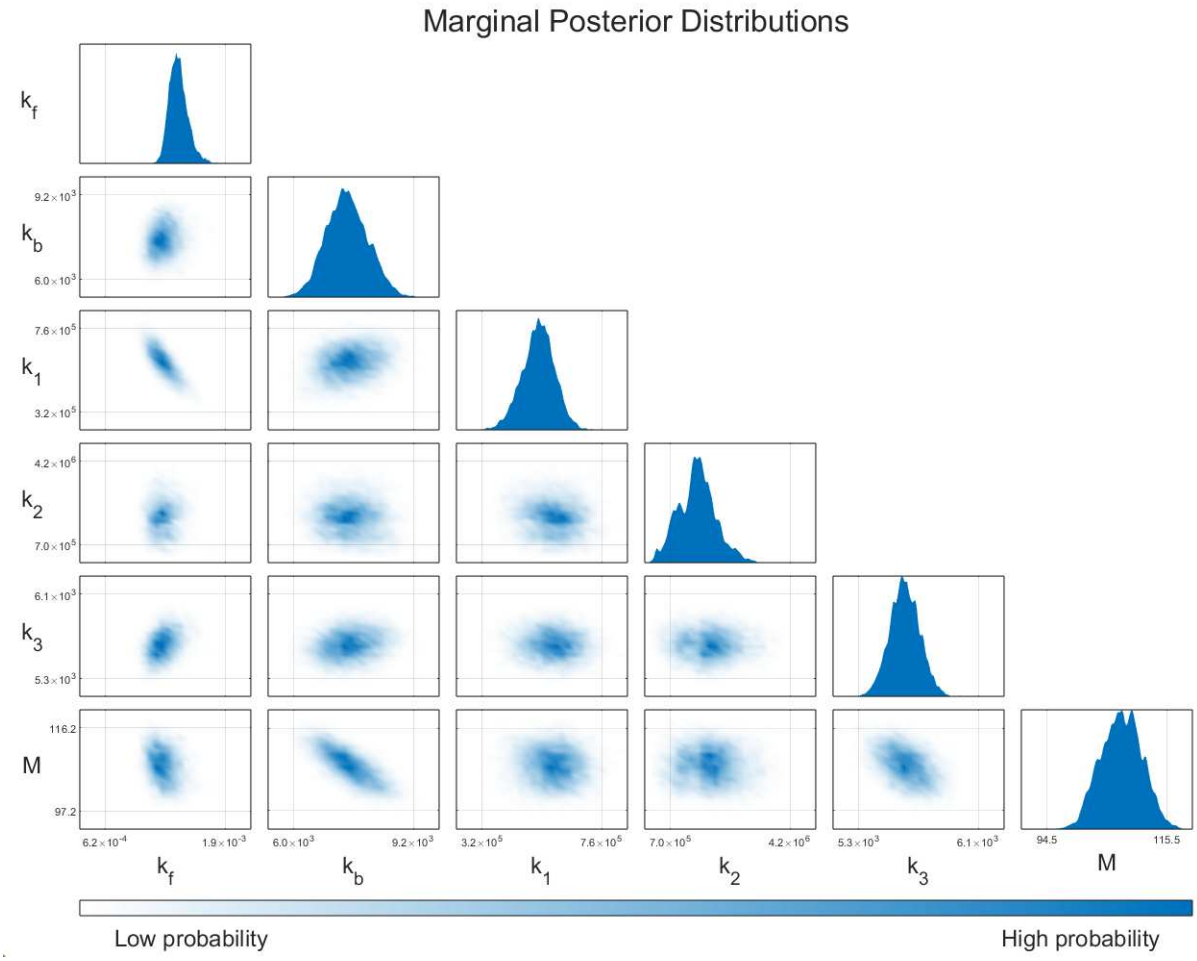


Figure 3.4: *1- and 2-dimensional marginal distributions for MCMC analysis B of the 3-step mechanism using the Metropolis-Hastings algorithm. The diagonal shows the 1-dimensional marginals and the lower-triangular regions shows the 2-dimensional marginal as indicated by the row and column labels. Summary statistics are presented in Table 3.3.*

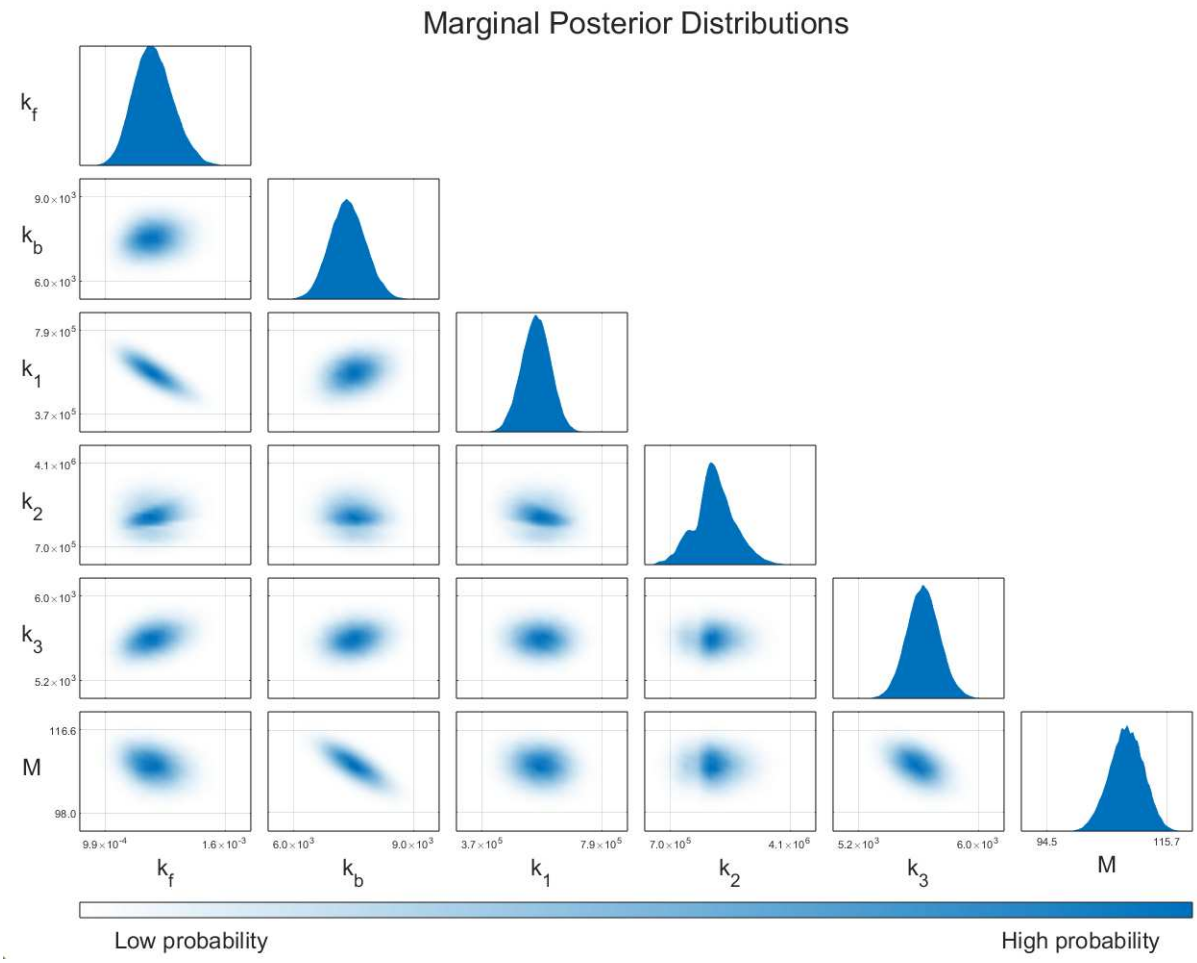


Figure 3.5: *1- and 2-dimensional marginal distributions for MCMC analysis C of the 3-step mechanism using the Metropolis-Hastings algorithm. The diagonal shows the 1-dimensional marginals and the lower-triangular regions shows the 2-dimensional marginal as indicated by the row and column labels. Summary statistics are presented in Table 3.4.*

3. The interesting correlation between k_1 and k_f has been smoothed out. In analysis A, a distinct "banana-shape" appeared, but the two dimensional marginal in Figure 3.5 shows a more simple oval. Strong correlation still exists, but it has been "normalized". This is likely due to the influence of using a normal approximation for the prior distribution. Future work could impose less structure via the prior distribution through use of a KDE or other methods to see if the "banana-shape" persists.

Next, I present my results for the 4-step mechanism.

Table 3.4: *Summary of the central tendencies of the posterior distribution for the MCMC analysis C of the 3-step mechanism using the Metropolis-Hastings algorithm. MAP is the maximum a posteriori, the maximum value observed for the posterior distribution. The posterior distribution is visualized in Figure 3.5.*

Parameter	MAP	Mean	$2.5\% - 97.5\%$ Quantile
k_f	1.2×10^{-3}	1.2×10^{-3}	$(1.1 - 1.4) \times 10^{-3}$
k_b	7.4×10^{3}	7.5×10^{3}	$(6.6 - 8.5) \times 10^{3}$
k_1	5.5×10^{5}	5.7×10^{5}	$(4.4 - 7.0) \times 10^{5}$
k_2	20.3×10^{5}	19.5×10^{5}	$(8.2 - 31.4) \times 10^{5}$
k_3	5.5×10^{3}	5.6×10^{3}	$(5.4 - 5.8) \times 10^{3}$
M	110	108	$102 - 114$

Table 3.5: *Summary of the central tendencies of the posterior distribution for MCMC analysis A of the 4-step mechanism using the Metropolis-Hastings algorithm. MAP is the maximum a posteriori, the maximum value observed for the posterior distribution. The posterior distribution is visualized in Figure 3.6.*

Parameter	MAP	Mean	$2.5\% - 97.5\%$ Quantile
k_f	2.3×10^{-3}	7.6×10^{-3}	$(2.4 - 58.0) \times 10^{-3}$
k_b	1.4×10^{4}	3.5×10^{4}	$(1.3 - 16.1) \times 10^{4}$
k_1	34.5×10^{4}	25.9×10^{4}	$(8.3 - 43.3) \times 10^{4}$
k_2	3.5×10^{4}	2.1×10^{4}	$(1.0 - 3.5) \times 10^{4}$
k_3	6.2×10^{3}	6.7×10^{3}	$(6.1 - 7.4) \times 10^{3}$
k_4	101.0×10^{1}	52.2×10^{1}	$(1.1 - 219.5) \times 10^{1}$
M	101	106	$90 - 134$

3.5.2 Inversion of the 4-step mechanism

For the 4-step mechanism, I go through the same three part analysis as with the 3-step mechanism. Analysis A is summarized with the marginal posterior distributions in Figure 3.6 and summary statistics in Table 3.5. My observations are:

1. Parameters k_1 and k_2 have broad distributions. They have a few modes each, but these modes are not in distinct regions. This indicates to me that more samples would blend the modes together to form a unimodal distribution.

2. Recall that k_4 is about agglomeration and is the additional physics added to the 3-step mechanism – i.e. the 3-step mechanism is the 4-step mechanism with $k_4 = 0$ – and thus part of my analysis is to investigate if adding this complexity to the model is significant. Here I note that the distribution of k_4 is broad, spanning three orders of magnitude.

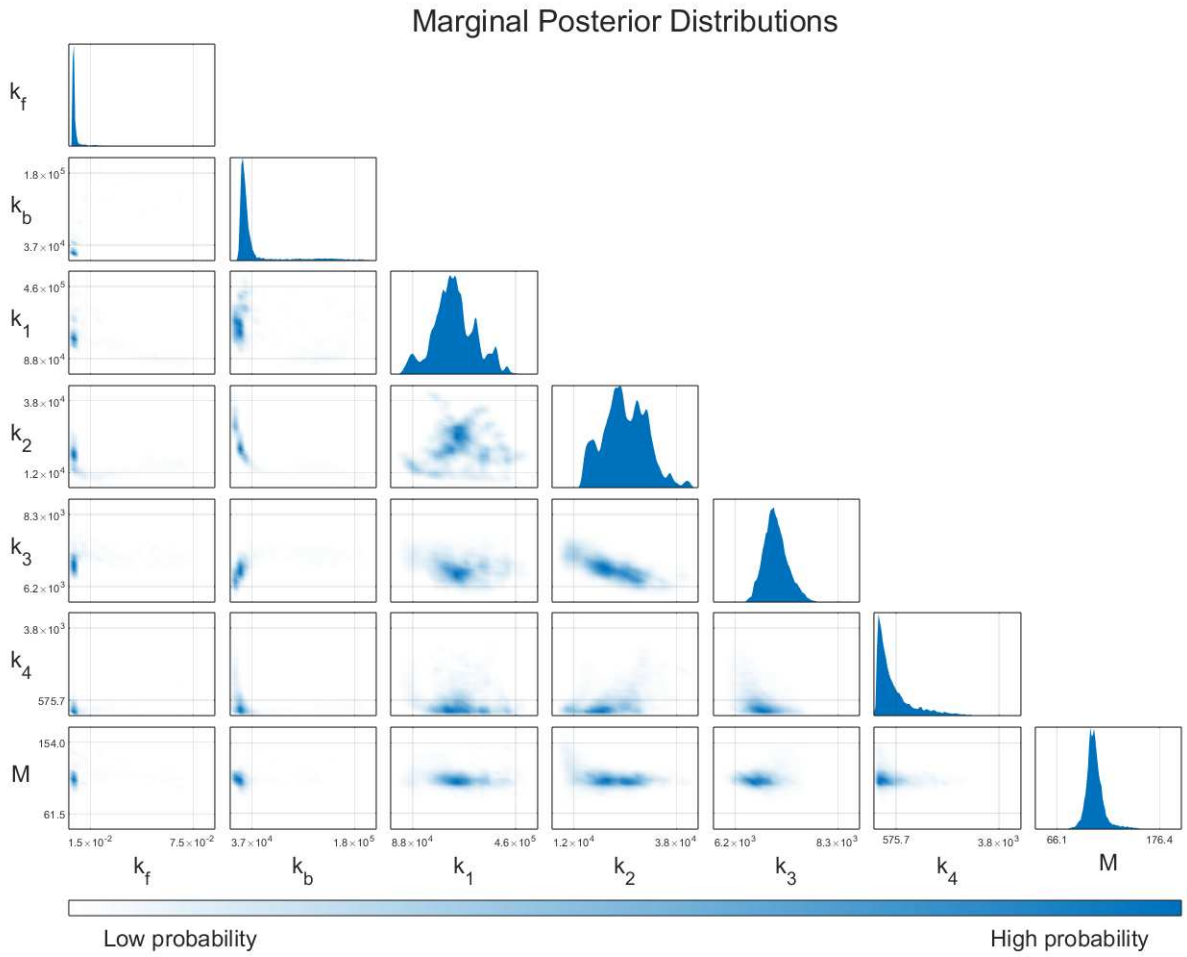


Figure 3.6: *1- and 2-dimensional marginal distributions for MCMC analysis A of the 4-step mechanism using the Metropolis-Hastings algorithm. The diagonal shows the 1-dimensional marginals and the lower-triangular regions shows the 2-dimensional marginal as indicated by the row and column labels. Summary statistics are presented in Table 3.5.*

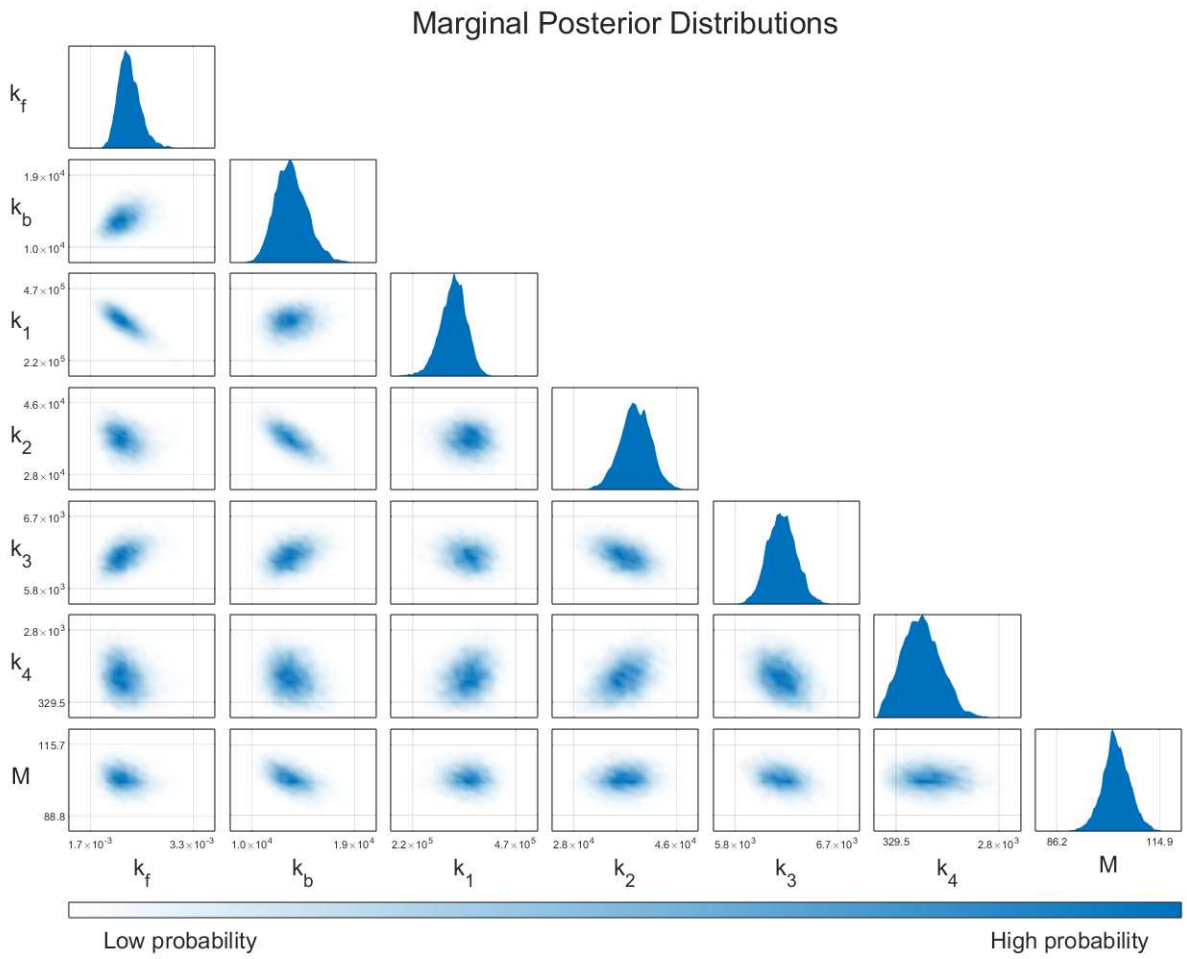


Figure 3.7: *1- and 2-dimensional marginal distributions for MCMC analysis B of the 4-step mechanism using the Metropolis-Hastings algorithm. The diagonal shows the 1-dimensional marginals and the lower-triangular regions shows the 2-dimensional marginal as indicated by the row and column labels. Summary statistics are presented in Table 3.6.*

Next, I conduct analysis B of the 4-step mechanism. The marginal posterior distributions are shown in Figure 3.7 and summary statistics of each parameter are in Table 3.6. I observe that:

1. Parameters k_1 and k_2 have settled into a much smaller region of parameter space. Their credible intervals no longer span two orders of magnitude.

2. Parameter k_4 still has a broad distribution.

Finally, I perform analysis C of the 4-step mechanism. The marginal posterior distributions are shown in Figure 3.8 and summary statistics of each parameter are in Table 3.7. I observe that:

Table 3.6: *Summary of the central tendencies of the posterior distribution for MCMC analysis B of the 4-step mechanism using the Metropolis-Hastings algorithm. MAP is the maximum a posteriori, the maximum value observed for the posterior distribution. The posterior distribution is visualized in Figure 3.7.*

Parameter	MAP	Mean	$2.5\% - 97.5\%$ Quantile
k_f	2.1×10^{-3}	2.2×10^{-3}	$(1.9 - 2.8) \times 10^{-3}$
k_b	1.4×10^{4}	1.4×10^{4}	$(1.1 - 1.7) \times 10^{4}$
k_1	3.6×10^{5}	3.6×10^{5}	$(2.7 - 4.3) \times 10^{5}$
k_2	3.7×10^{4}	3.7×10^{4}	$(3.1 - 4.3) \times 10^{4}$
k_3	6.2×10^{3}	6.2×10^{3}	$(5.9 - 6.5) \times 10^{3}$
k_4	9.6×10^{2}	11.9×10^{2}	$(2.1 - 22.8) \times 10^{2}$
M	102	102	$95 - 110$

Table 3.7: *Summary of the central tendencies of the posterior distribution for MCMC analysis C of the 4-step mechanism using the Metropolis-Hastings algorithm. MAP is the maximum a posteriori, the maximum value observed for the posterior distribution. The posterior distribution is visualized in Figure 3.8.*

Parameter	MAP	Mean	$2.5\% - 97.5\%$ Quantile
k_f	2.0×10^{-3}	2.1×10^{-3}	$(1.8 - 2.4) \times 10^{-3}$
k_b	1.3×10^{4}	1.3×10^{4}	$(1.1 - 1.5) \times 10^{4}$
k_1	4.0×10^{5}	3.9×10^{5}	$(3.2 - 4.5) \times 10^{5}$
k_2	4.0×10^{4}	4.0×10^{4}	$(3.4 - 4.6) \times 10^{4}$
k_3	6.1×10^{3}	6.1×10^{3}	$(5.9 - 6.4) \times 10^{3}$
k_4	15.2×10^{2}	13.4×10^{2}	$(3.8 - 23.5) \times 10^{2}$
M	102	103	$97 - 109$

1. The parameter k_b takes on values on the order of 10^4 whereas in the 3-step mechanism values for k_b on the order of 10^3 are found. That being said, the *ratio* k_f/k_b is similar: 1.5×10^{-7} for the 4-step and 1.6×10^{-7} for the 3-step.

2. The parameter k_2 takes values on the order of 10^4 (with tight uncertainty bounds) in the 4-step mechanism, whereas the 3-step mechanism finds values of $10^5 - 10^6$ for k_2. Of course, smaller particles have their growth affected by both parameters k_2 and k_4 in the 4-step model while the 3-step model has small particle growth only affected by k_2. Since k_4 has large uncertainty in the 4-step, it appears as if the uncertainty present in k_2 in the 3-step model has been transferred to k_4 in the 4-step model. In other words, my interpretation is that adding agglomeration to the 4-step model has not necessarily reduced the uncertainty in the model but, rather, has just moved it to a different parameter.

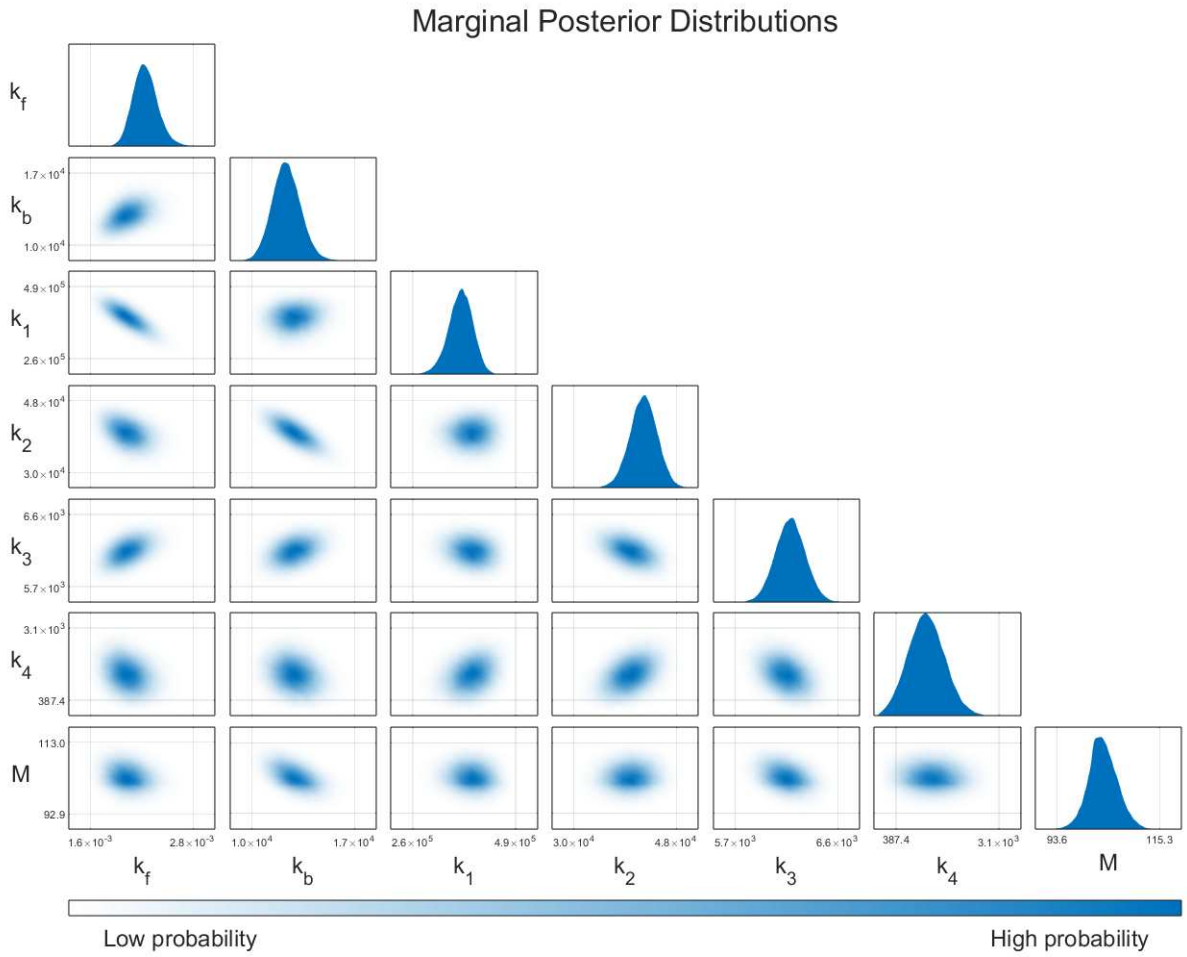


Figure 3.8: *1- and 2-dimensional marginal distributions for MCMC analysis C of the 4-step mechanism using the Metropolis-Hastings algorithm. The diagonal shows the 1-dimensional marginals and the lower-triangular regions shows the 2-dimensional marginal as indicated by the row and column labels. Summary statistics are presented in Table 3.7.*

I performed Bayesian inversion analyses for both the 3-step and 4-step mechanisms. The goal of these analyses is to figure model parameters that correspond to simulations that fit well to the data. Hence, it is important to visualize my results not only in terms of the parameter spaces Figure 3.5 and Figure 3.8, but also in terms of what I *expect* a measurement to yield based on my simulations. The next section discusses this visualization.

3.5.3 Data fits

In previous work with my collaborators, we have visualized a simulation fitting to the data in the manner of Figure 2.1. However, with how noisy the data are, I found it challenging to visually assess how well a simulation matched the data. Therefore, I developed a better way to visualize the simulation so that it compared to the data in a more direct way. The idea is to perform a simulation of the *measurement process*, not simply the concentrations over time. The measurement process is akin to drawing from a multinomial distribution. Therefore, to better visualize the simulations versus data, I do the following:

1. Draw 10000 samples from the posterior distribution.

2. Solve the ODEs using the posterior samples to compute concentrations for each particle at each time when data were collected.

3. Convert particle concentrations into probabilities of "drawing" that particle according to (3.3).

4. Simulate data collection by drawing an equal number of total particles from the simulated multinomial distribution as was observed in the actual data.

5. Graph the distribution of the simulated measurements alongside the observed data. These are organized in $0.1\,\mathrm{nm}$ bins.

The first data were observed after 0.918 hours. Figure 3.9 shows the fits of the 3-step mechanism (top) and the 4-step mechanism (bottom). Note that these visualizations include the uncertainty in the parameter estimates due to how the visualization was constructed. A "good fit" can be

determined by if the data – represented by a black circle in the figures I present herein – is located in a high probability region. For this first set of data, the fits are quite similar between the 3-step and 4-step mechanisms. The only notable difference I find is the data in the $1.6\,\mathrm{nm}$ to $1.7\,\mathrm{nm}$ range is much more likely to be observed in the 3-step mechanism.

The second data are observed after 1.170 hours and are visualized in Figure 3.10. Note that this data is very sparse, only containing 61 particle observations whereas the first, third, and fourth data sets contain 246, 150, and 213 particle observations respectively. Overall, I find the fits to be approximately the same. The observed data for bins $1.5\,\mathrm{nm}$ to $1.6\,\mathrm{nm}$, $2.8\,\mathrm{nm}$ to $2.9\,\mathrm{nm}$, $2.9\,\mathrm{nm}$ to $3.0\,\mathrm{nm}$, and $3.8\,\mathrm{nm}$ to $3.9\,\mathrm{nm}$ seem to be especially unlikely according to both mechanisms.

The third data are observed after 2.336 hours and I show the simulation versus data in Figure 3.11. Again, I find the fits to be very similar. The observations in bins $2.3\,\mathrm{nm}$ to $2.4\,\mathrm{nm}$ and $2.6\,\mathrm{nm}$ to $2.7\,\mathrm{nm}$ are notably unlikely according to my simulations.

The final data are observed after 4.838 hours and this marks the end of the reaction. From the visualization in Figure 3.12, I again conclude that, visually, the simulation fits the data similarly between the 3-step and 4-step mechanism. There appear to be outlier observations in bins $2.8\,\mathrm{nm}$ to $2.9\,\mathrm{nm}$ and $3.1\,\mathrm{nm}$ to $3.2\,\mathrm{nm}$ but overall the simulations agree well with the data.

The visualizations in this section show agreement between both mechanisms and the observed data. In other words, the collected data is reasonably within the uncertainty stemming from the distributions I constructed during my Bayesian inversion analyses. However, these visualizations *assume* that I found the posterior distribution. In the next section, I conduct detailed analyses of the convergence of my posterior distributions so that I am confident in my results.

3.6 Chain convergence

In this section, I will talk about the determination of whether the MCMC posterior distributions in Section 3.5 have converged to the posterior distribution or not. For both the 3-step and 4-step mechanisms, I find that:

- Analysis A does not converge to the posterior distribution;

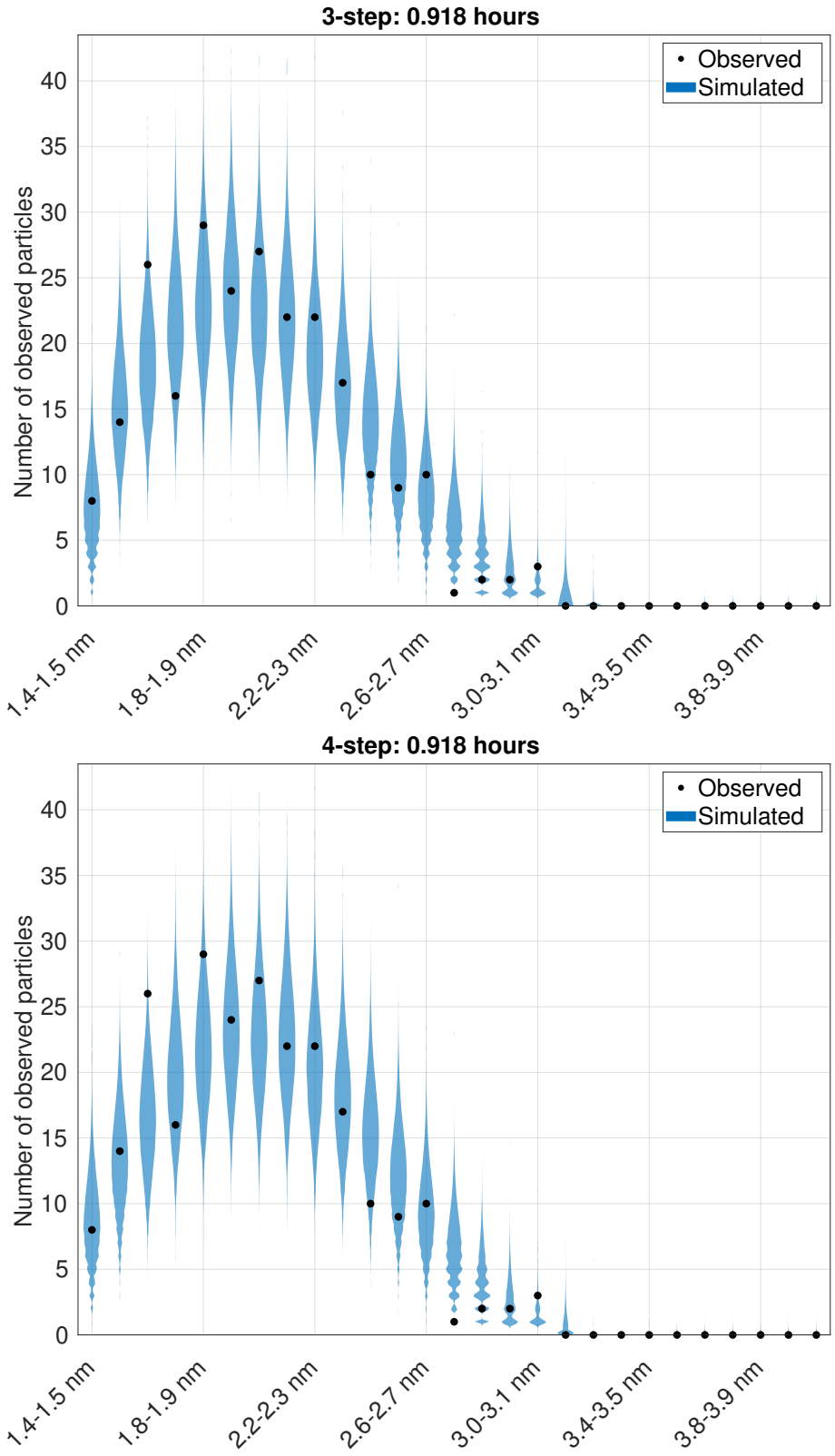


Figure 3.9: *Observed data in red plotted against simulated data in blue of (top) the 3-step mechanism and (bottom) the 4-step mechanism* 0.918 *hours into the reaction. A good fit is determined by the observed data being in a high probability region of the simulated data.*

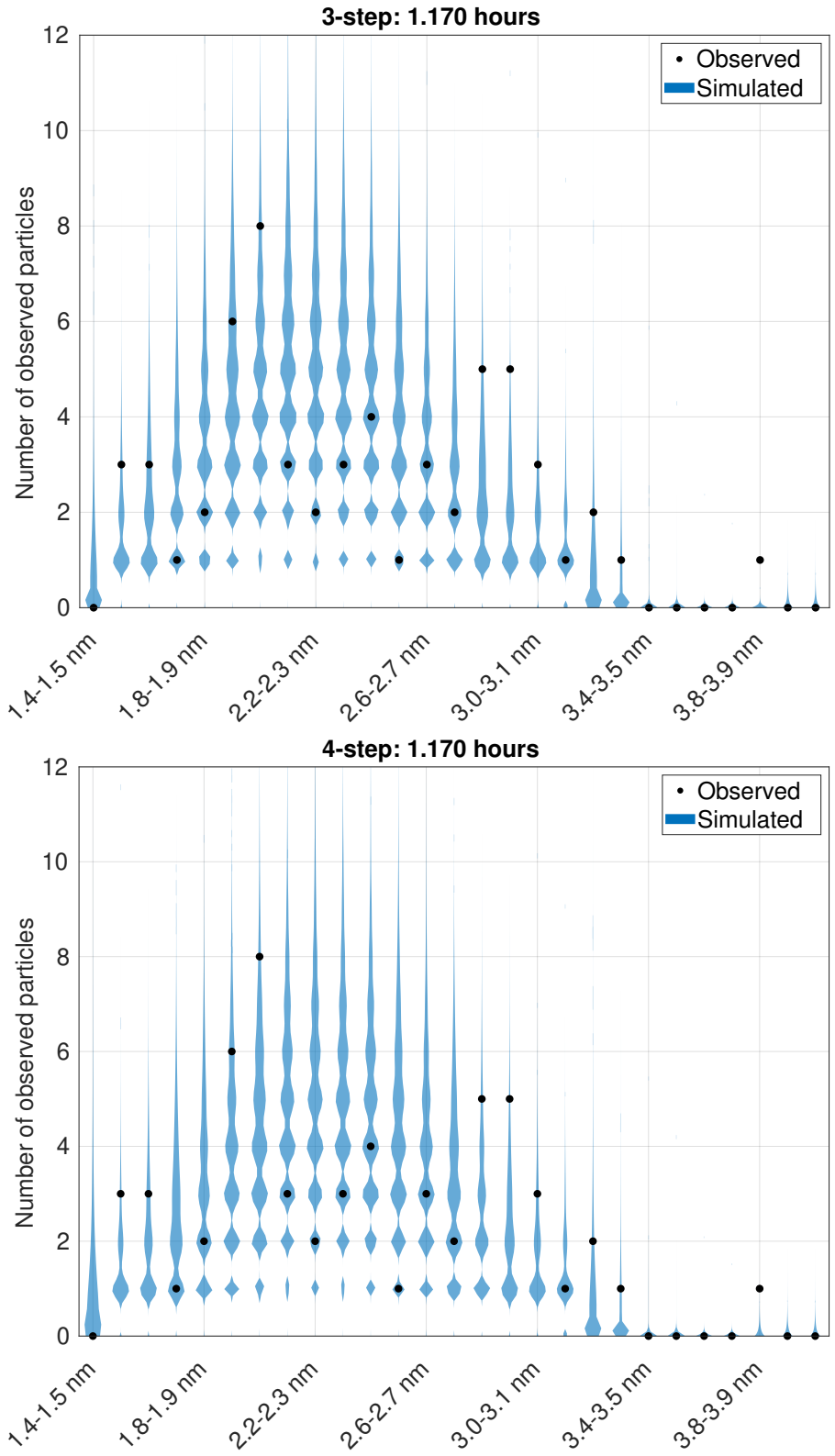


Figure 3.10: *Observed data in red plotted against simulated data in blue of (top) the 3-step mechanism and (bottom) the 4-step mechanism* 1.170 *hours into the reaction. A good fit is determined by the observed data being in a high probability region of the simulated data.*

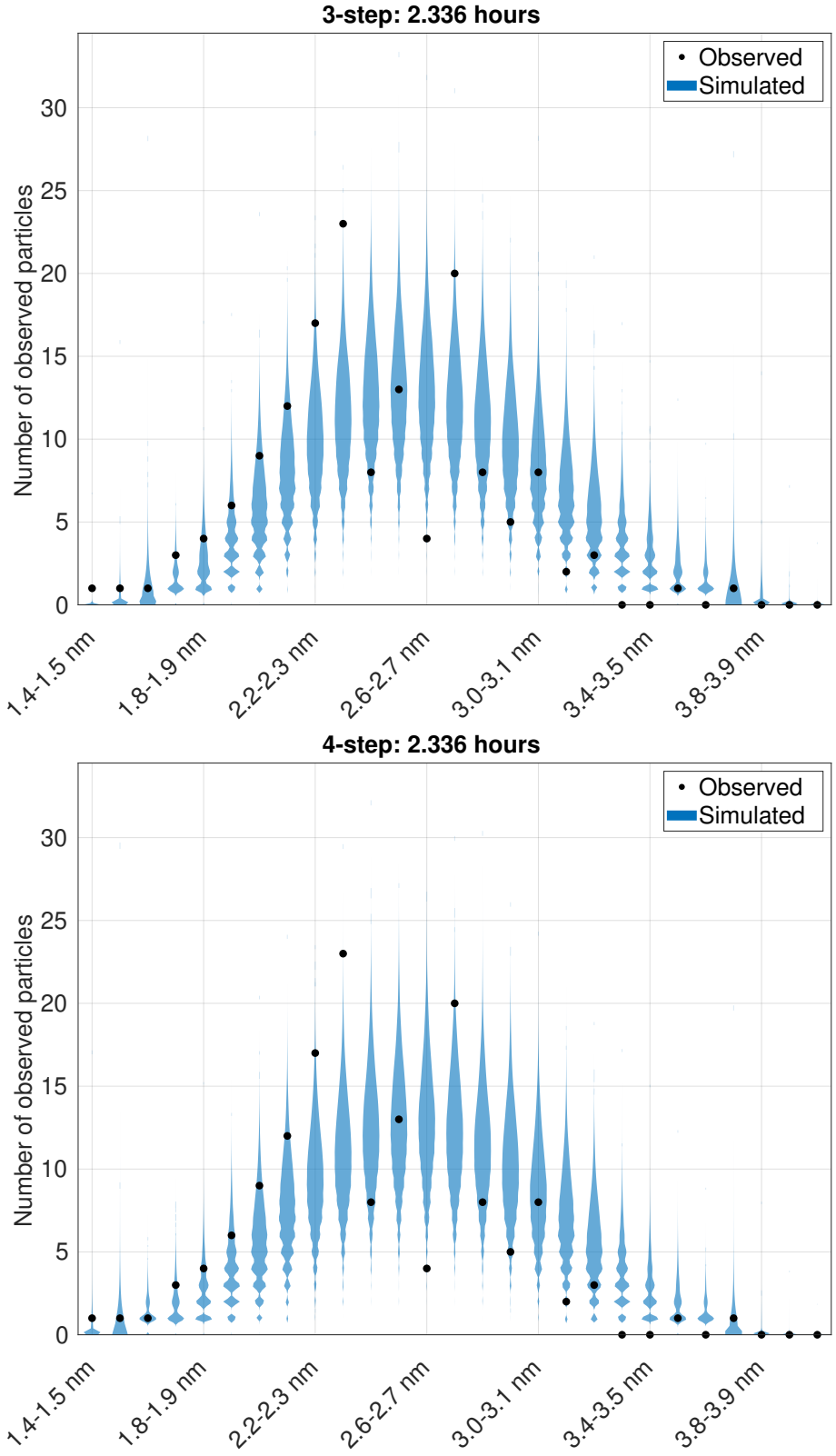


Figure 3.11: *Observed data in red plotted against simulated data in blue of (top) the 3-step mechanism and (bottom) the 4-step mechanism* 2.336 *hours into the reaction. A good fit is determined by the observed data being in a high probability region of the simulated data.*

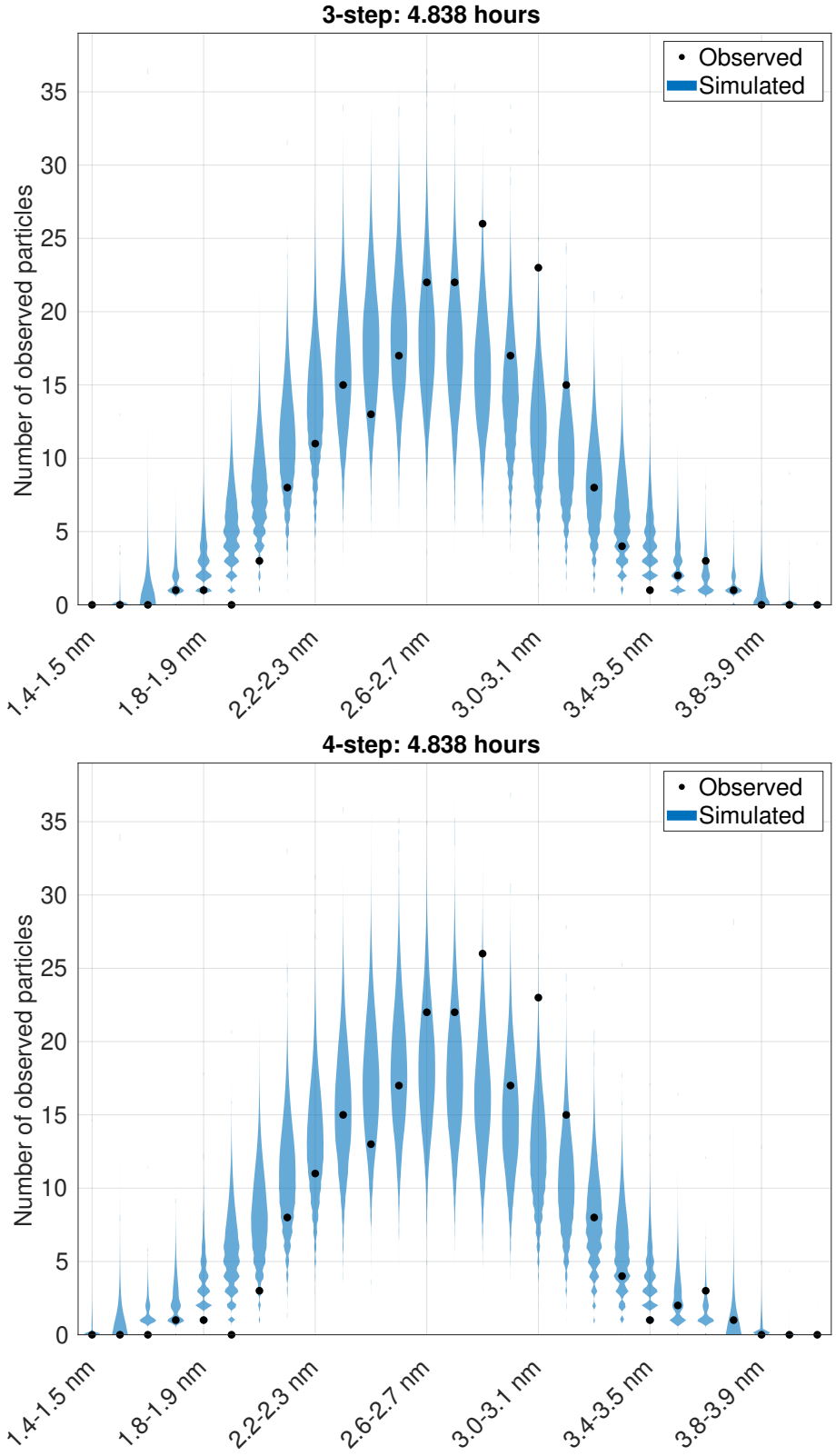


Figure 3.12: *Observed data in red plotted against simulated data in blue of (top) the 3-step mechanism and (bottom) the 4-step mechanism* 4.838 *hours into the reaction. A good fit is determined by the observed data being in a high probability region of the simulated data.*

- Analysis B does appear to have converged, but has fewer independent samples than I want;
- Analysis C has converged to the posterior and I am comfortable using this analysis for final results.

This section goes through my process of coming to these conclusions. In order to do this, my goals in this section are to:

1. Assess the convergence of my posterior distributions to gain confidence in my results.
2. Reduce the autocorrelation in my chains so that techniques such as Monte Carlo integration do not suffer from slower convergence.

Note that for the second goal I will perform a process called "thinning" – a process by which every kth sample is discarded – but I do not actually perform this during this chapter. For figures such as Figure 3.8 the difference in computational expense is negligible. For figures such as Figure 3.12, there certainly is a difference in computational expense between solving, say, 10000 ODEs versus 1000 ODEs, but I found the time difference to be on the scale of minutes, rather than hours or days, so I found it to be acceptable to not thin my samples. On the other hand, for the optimization procedures performed in the subsequent chapters, I will have to solve thousands of ODEs every iteration of an iterative algorithm. In that case, it does matter how many times I need to solve my ODEs each iteration and thus I thin my samples for Chapter 4 and Chapter 5.

Assessing convergence for an MCMC analysis is a heuristic process. Therefore it is helpful to look at convergence through a few of these heuristics before making a decision. The most intuitive way to assess convergence is by considering trace plots, which plot the sample number in an MCMC chain on the x-axis and the parameter value on the y-axis. Figure 3.13 demonstrates some examples of what this looks like. The leftmost plot shows what a converged chain typically looks like: centered around some mean value with a constant variance around the mean. In this case, I would say the chain has reached a *stationary distribution*. Informally, a stationary distribution refers to when a chain has reached a state where its random walk is centered around a common mean value and its variance is fixed; see [76, section 3.3] for a formal discussion. The left-center

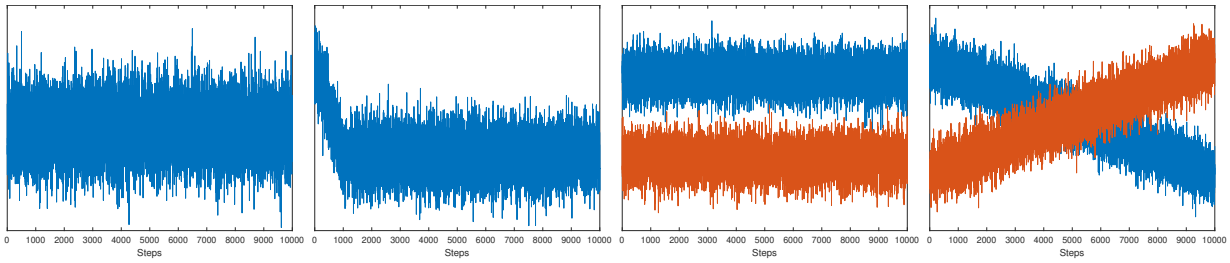


Figure 3.13: *(Left) Example of a trace plot for a converged MCMC chain. (Left-center) Example of a trace plot for a converged MCMC chain after a burn-in period. (Right-center) Example of unconverged MCMC chains when two chains have not settled on the same distribution. (Right) Example of unconverged MCMC chains that have not settled into stationary distributions.*

plot shows a typical phenomenon wherein a chain takes a number of steps traveling to a different region of parameter-space before settling into a stationary distribution. This phenomenon is typically dealt with via applying a "burn-in" period where the first K samples are discarded [35]. The right-center plot shows an example of two MCMC chains that, separately, seem to have reached stationary distributions, but they do not overlap. This behavior either indicates (1) that the chains are not converged and will join together if they are run sufficiently long, or (2) that the chains follow a multimodal distribution and are "stuck" in separate modes, although as the number of steps grows large the chains would travel to the other modes. Finally, the rightmost plot of Figure 3.13 shows an extreme case where two chains are not stationary. This example will also be referred to when I discuss other convergence assessments later in this section.

I generate trace plots for every parameter of the 3-step and 4-step mechanisms. For analysis A of each mechanism I produce Figure 3.14 and Figure 3.15. For the plots of the 3-step mechanism in Figure 3.14, there are very clear signs of non-convergence. Parameters k_1 and k_2 have not settled to a stationary distribution for individual chains or as a collective. Parameters k_f, k_b, and k_3 appear to still be traveling to a higher probability region of parameter space. Parameter M does appear to be stationary, however. The results I show in Figure 3.15 tell a similar story, although more extreme. Parameters k_1 and k_2 have not found agreement. Parameters k_f, k_b, k_3, and k_4 appear to be moving towards a stationary distribution, although k_4 also shows significant variance even near the end of the chain. Parameter M appears to have found a stationary distribution after approximately 50% of the steps.

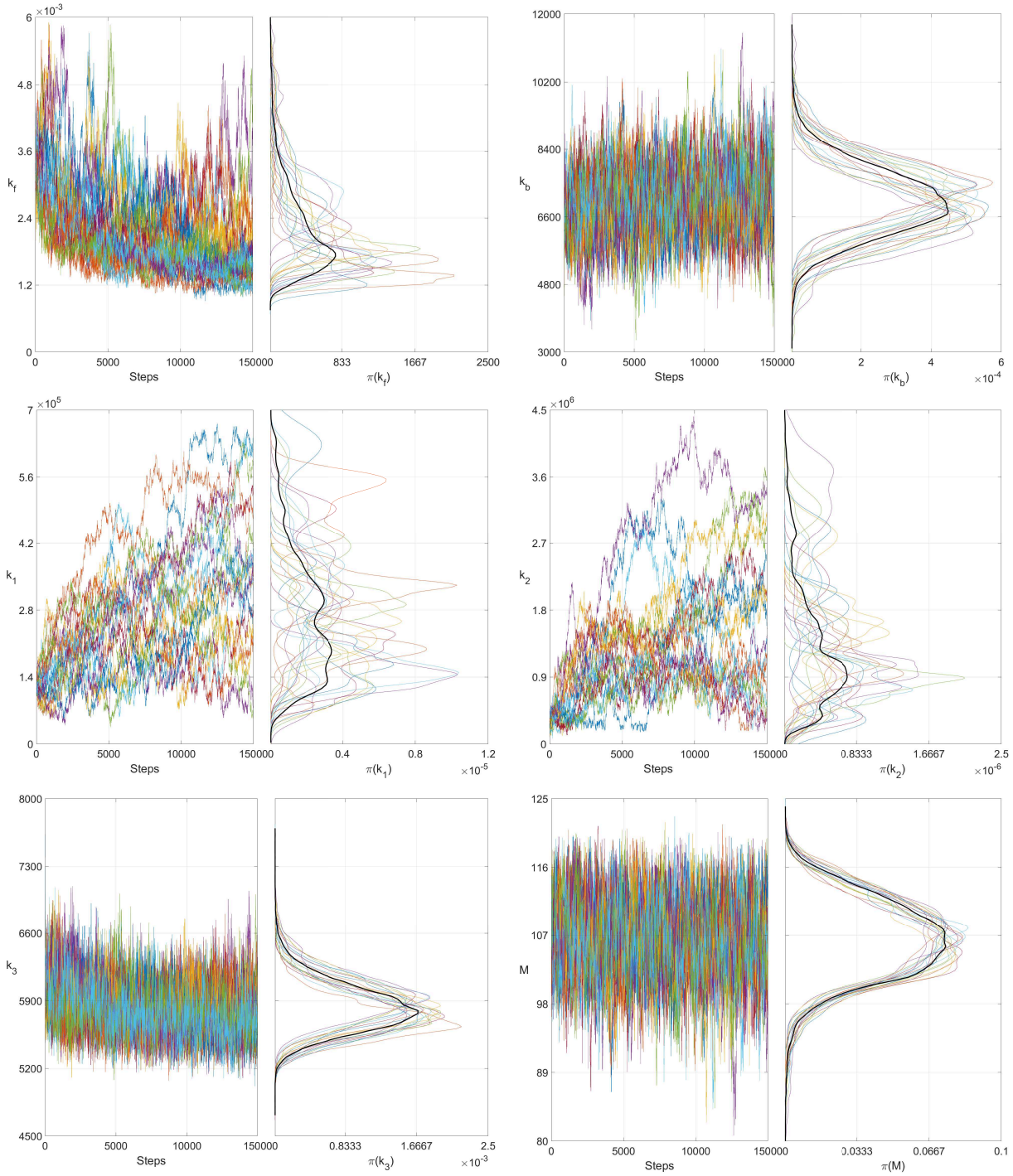


Figure 3.14: *Trace plots for the first MCMC analysis of the 3-step mechanism. In each plot, the left subplot shows the parameter value plotted alongside the sample number with each individual chain plotted in a different color. The right subplot shows a KDE of the marginal distribution for each chain in separate colors and for all chains together in black.*

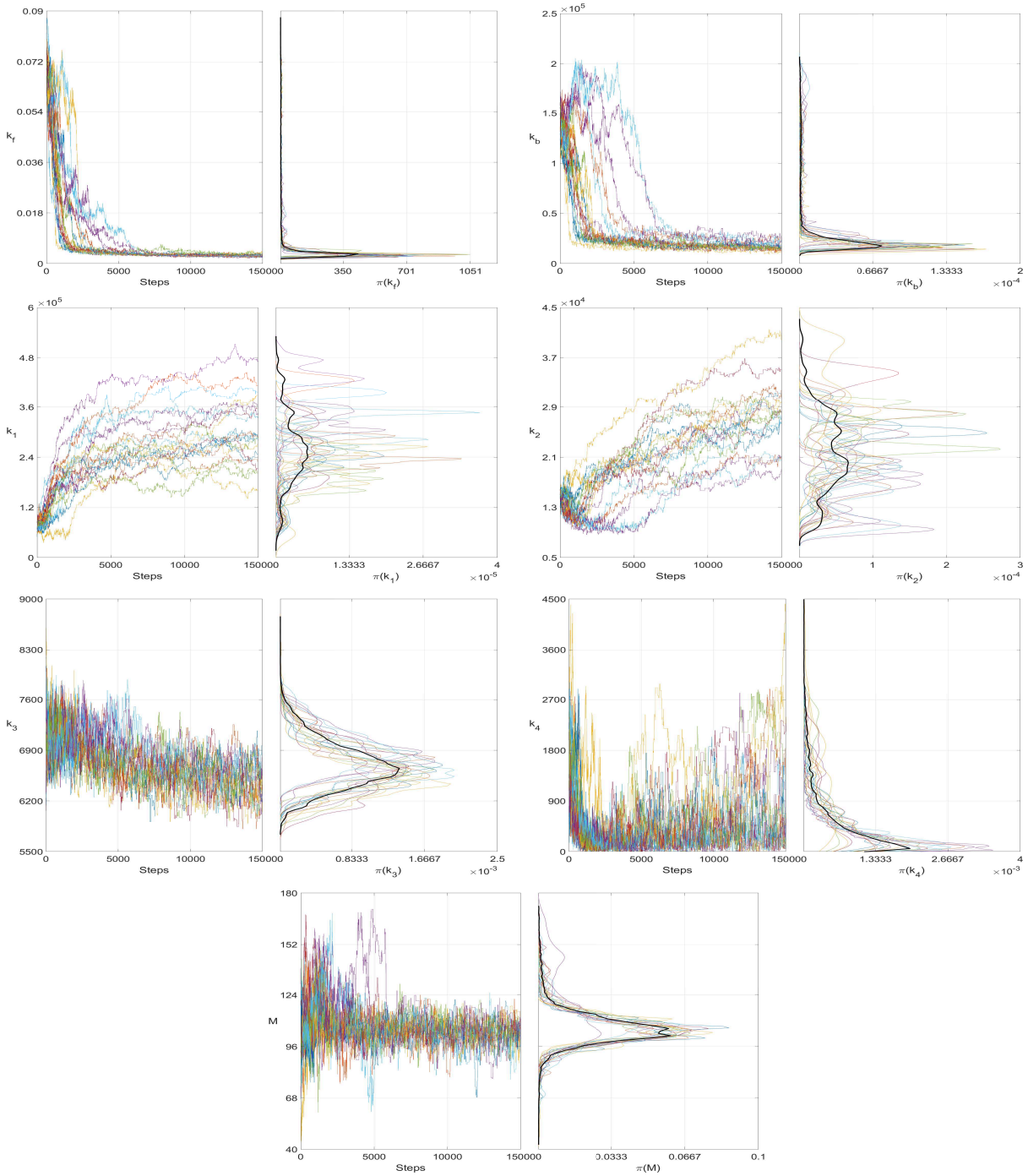


Figure 3.15: *Trace plots for the first MCMC analysis of the 4-step mechanism. In each plot, the left subplot shows the parameter value plotted alongside the sample number with each individual chain plotted in a different color. The right subplot shows a KDE of the marginal distribution for each chain in separate colors and for all chains together in black.*

Next, Figure 3.16 and Figure 3.17 show the trace plots for analysis B for each mechanism. The results for both analyses are similar so I will describe my interpretations together. For most of the parameters, there appears to be a small period with higher variance where the chain travels (in its mean value) slightly. This lasts approximately 1000 samples before settling into what appears to be a fairly stationary distribution. That being said, the KDE I include on these plots shows substantial roughness – that is, the "bumpiness" – and this is indicative of chains not overlapping a lot. In this case, this is because both of these analyses had a very low acceptance rate for their chains, as can be seen in Figure 3.20. I discuss this further at the end of this section, but this low acceptance ratio is what prompted me to conduct the third round of MCMC.

Finally, Figure 3.18 and Figure 3.19 show the trace plots for the final round of MCMC analysis. For all parameters the parameters appear to be varying about the same mean for every chain. Moreover, the KDE for individual chains are quite similar to the KDE for all combined chains. There also does not appear to be any burn-in period. I consider these distributions converged based on the trace plots.

The next convergence heuristic I discuss is a quantitative assessment of convergence. Although this is a numerical score rather than a qualitative deduction, interpreting what the score means is still a subjective measure and thus this is still a heuristic. The intuition behind this score is that if multiple chains have converged to a stationary distribution, then the variation exhibited within the chains and between the chains should be comparable. This convergence score is presented in [35] and the following equations are reproduced from this source. Let $\mathcal{P}_i^{(j)}$ be a scalar value corresponding to the ith sample of the jth chain. For the purposes of my work, I interpret this as a single parameter value since I am interested in the posterior distribution of the parameter space, but if one was interested in the posterior distribution of some arbitrary quantity of interest, then

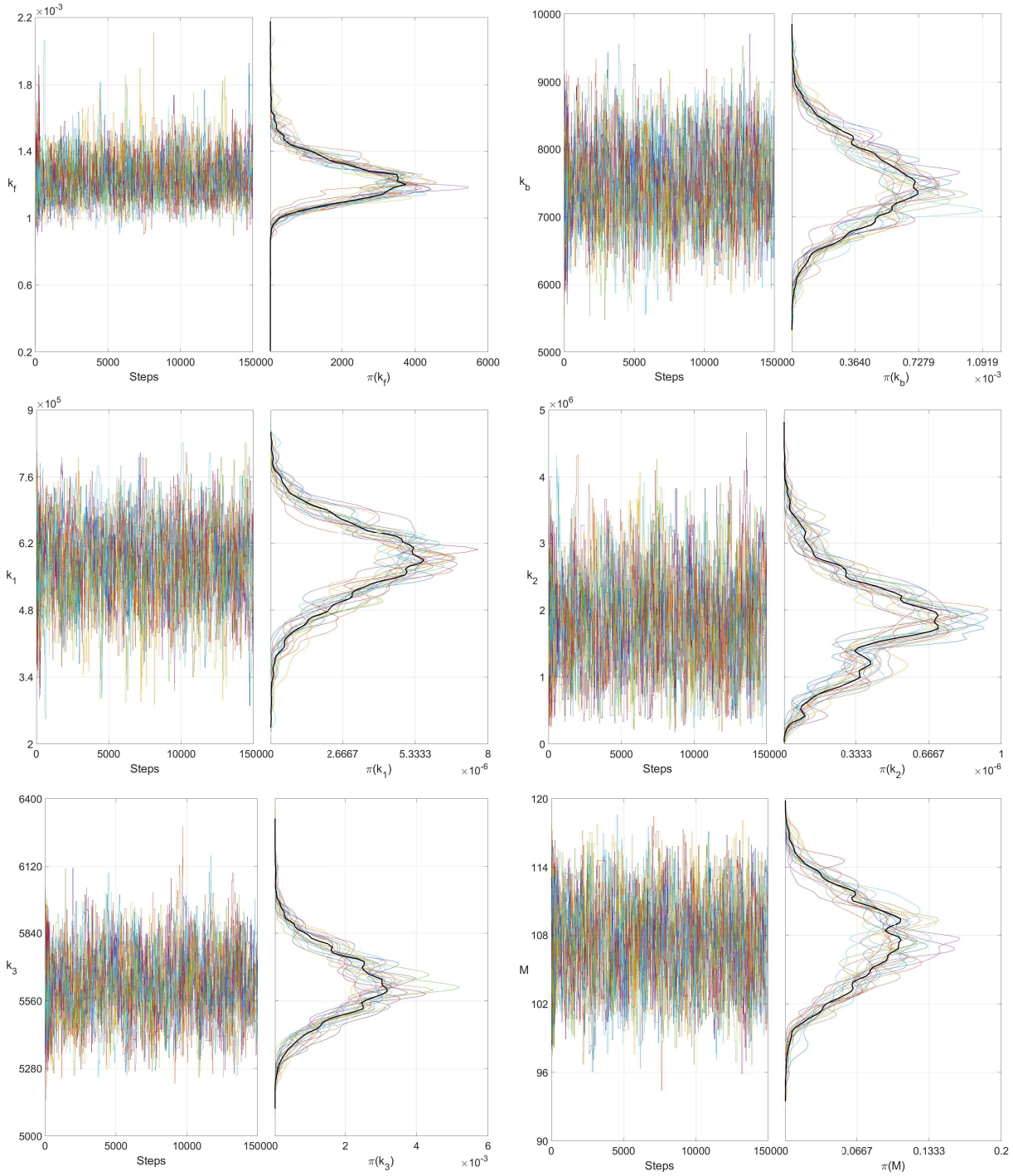


Figure 3.16: *Trace plots for the second MCMC analysis of the 3-step mechanism. In each plot, the left subplot shows the parameter value plotted alongside the sample number with each individual chain plotted in a different color. The right subplot shows a KDE of the marginal distribution for each chain in separate colors and for all chains together in black.*

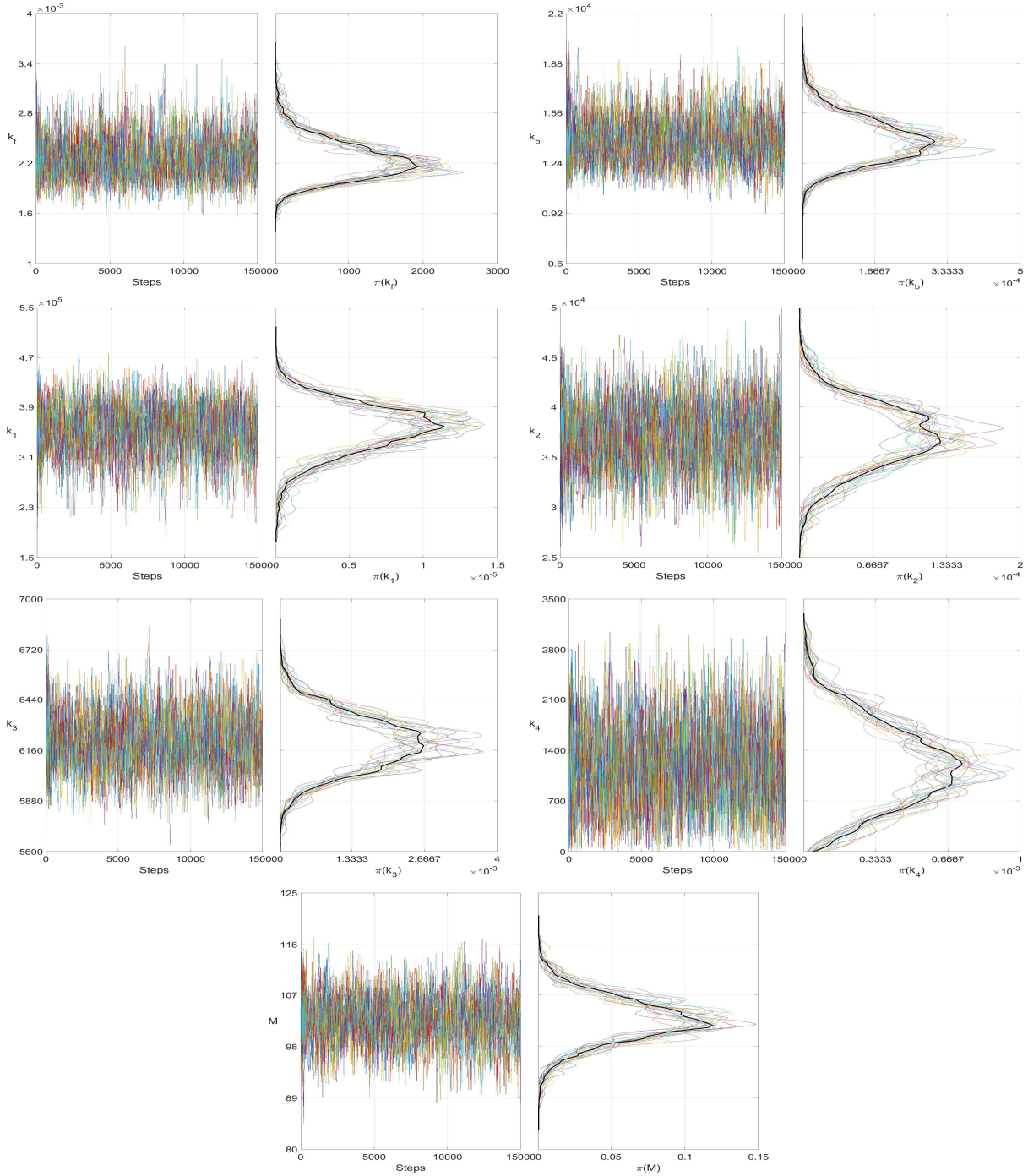


Figure 3.17: *Trace plots for the second MCMC analysis of the 4-step mechanism. In each plot, the left subplot shows the parameter value plotted alongside the sample number with each individual chain plotted in a different color. The right subplot shows a KDE of the marginal distribution for each chain in separate colors and for all chains together in black.*

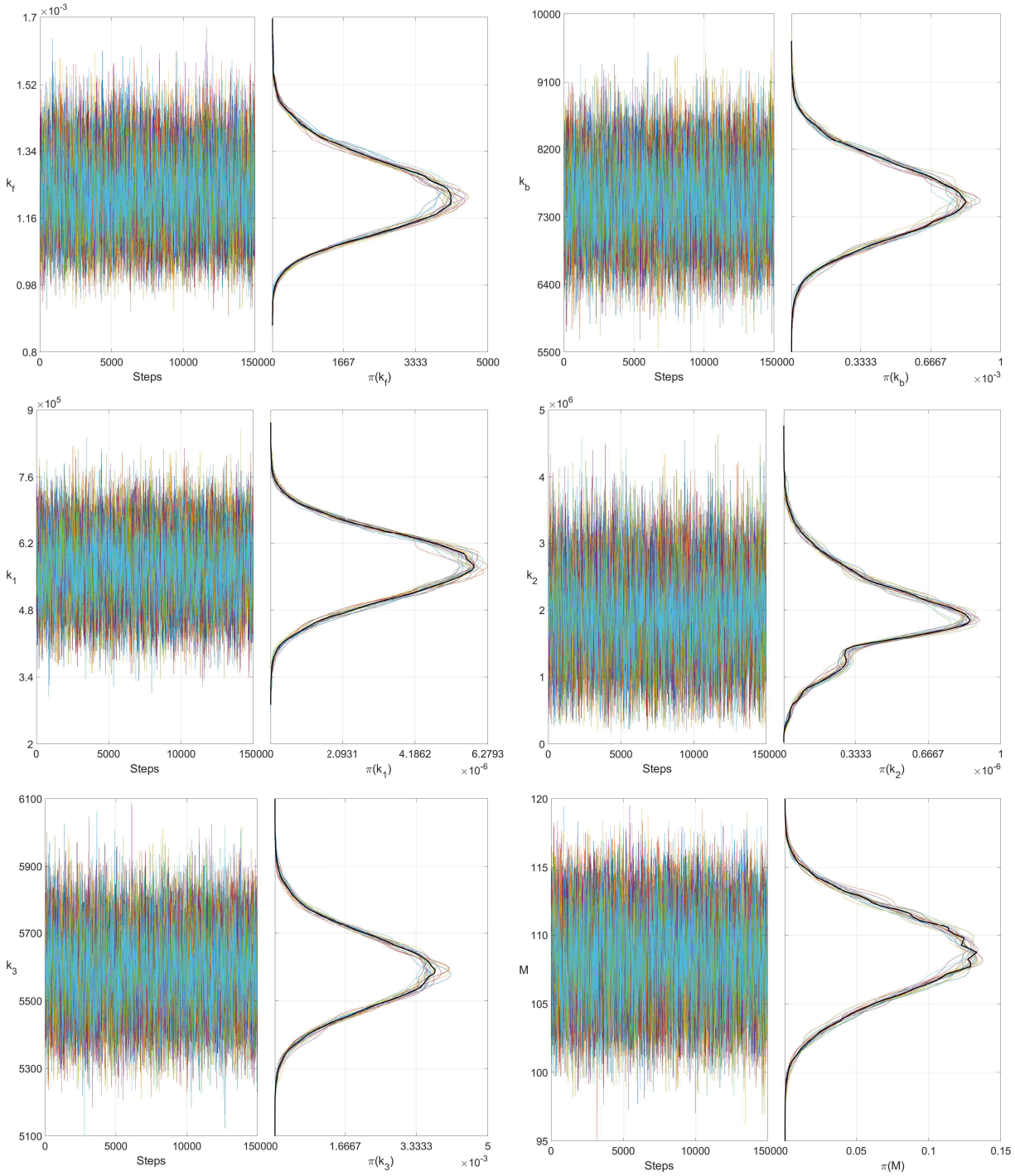


Figure 3.18: *Trace plots for the third MCMC analysis of the 3-step mechanism. In each plot, the left subplot shows the parameter value plotted alongside the sample number with each individual chain plotted in a different color. The right subplot shows a KDE of the marginal distribution for each chain in separate colors and for all chains together in black.*

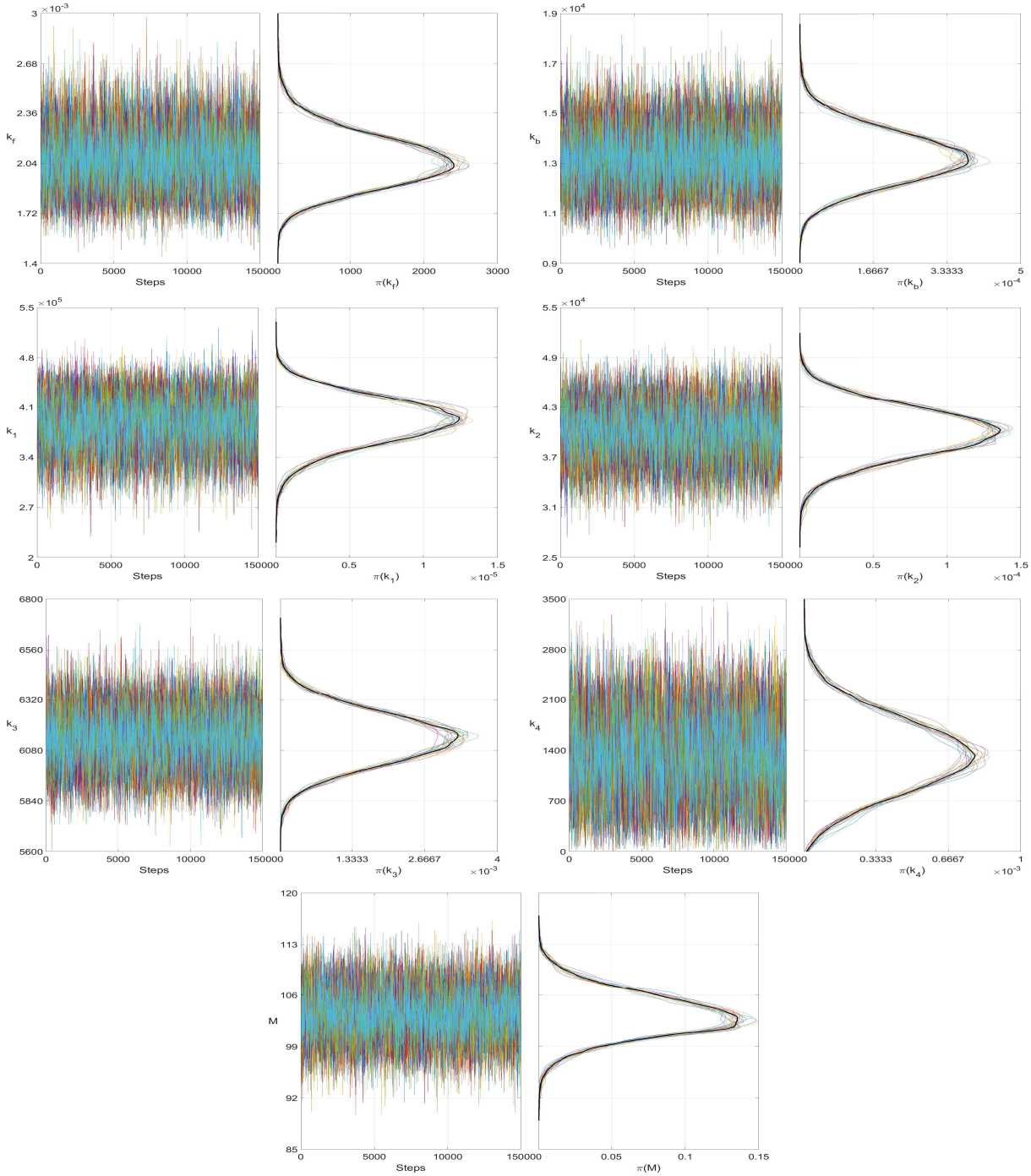


Figure 3.19: *Trace plots for the third MCMC analysis of the 4-step mechanism. In each plot, the left subplot shows the parameter value plotted alongside the sample number with each individual chain plotted in a different color. The right subplot shows a KDE of the marginal distribution for each chain in separate colors and for all chains together in black.*

this same analysis holds. Then, if there are M chains and N samples per chain, I compute [1]

$$B = \frac{1}{M-1}\sum_{j=1}^{M}\left(\bar{\mathcal{P}}^{(j)} - \bar{\mathcal{P}}\right)^2, \quad \text{where} \quad \bar{\mathcal{P}}^{(j)} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{P}_i^{(j)}, \quad \bar{\mathcal{P}} = \frac{1}{M}\sum_{j=1}^{M}\bar{\mathcal{P}}^{(j)} \tag{3.25}$$

$$W = \frac{1}{M}\sum_{j=1}^{M}s_j^2, \quad \text{where} \quad s_j^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(\mathcal{P}_i^{(j)} - \bar{\mathcal{P}}^{(j)}\right)^2. \tag{3.26}$$

From (3.25) and (3.26), an *overestimate* of the variance posterior distribution is constructed with

$$\widehat{\text{var}}^{+} = \frac{N-1}{N}W + B. \tag{3.27}$$

While $\widehat{\text{var}}^{+}$ is an overestimate of the variance, W is an underestimate of the variance. This is because W is the average variance of each chain, but since the chains contain finitely many samples they have not had time to explore the entire range of the posterior distribution. Thus, in [35], the convergence statistic to monitor is an estimate of the factor by which the variance of the posterior might reduce if the MCMC sampling was run in perpetuity. This statistic – called the *potential scale reduction factor* – is given by

$$\widehat{R} = \sqrt{\frac{\widehat{\text{var}}^{+}}{W}}. \tag{3.28}$$

$\widehat{R} \to 1$ as $N \to \infty$ and typically a threshold of $\widehat{R} < 1.1$ is taken as evidence of convergence.

A similar statistic has been developed to assess the convergence of all the parameters simultaneously. This is called the *multivariate potential scale reduction factor (MPSRF)* [13]. Let $\boldsymbol{\mathcal{P}}_i^{(j)}$ be the ith column-vector of parameters in the jth chain. Then let N be the number of samples per

[1] In the original text, what I call B – the between chain variance – is denoted as B/N and thus the calculations of (3.27) and (3.31) include this factor of $\frac{1}{N}$. I modify the notation in my work because I found the notation B/N to be cumbersome and confusing.

Table 3.8: *Convergence statistics for each MCMC analysis. The MPSRF column indicates* (3.31) *and considers all parameters simultaneously; a value less than 1.1 is typically considered an indication of convergence. The other columns indicate* (3.28) *calculated for the individual parameter; all parameters having values less than 1.1 is typically considered an indication of convergence.*

Analysis	k_f	k_b	k_1	k_2	k_3	k_4	M	MPSRF
3-Step A	1.414	1.109	1.834	1.696	1.099	–	1.016	2.362
3-Step B	1.023	1.015	1.027	1.023	1.014	–	1.014	1.041
3-Step C	1.001	1.002	1.001	1.001	1.001	–	1.002	1.002
4-Step A	1.110	1.393	1.781	2.127	1.310	1.236	1.136	7.896
4-Step B	1.014	1.011	1.015	1.010	1.014	1.012	1.013	1.032
4-Step C	1.001	1.002	1.002	1.002	1.001	1.002	1.002	1.005

chain and M be the number of chains. Then define[1]

$$\begin{aligned} \boldsymbol{B} &= \frac{1}{M-1}\sum_{j=1}^{M}\left(\bar{\boldsymbol{\mathcal{P}}}^{(j)}-\bar{\boldsymbol{\mathcal{P}}}\right)\left(\bar{\boldsymbol{\mathcal{P}}}^{(j)}-\bar{\boldsymbol{\mathcal{P}}}\right)^{\intercal}, \\ &\text{where} \quad \bar{\boldsymbol{\mathcal{P}}}^{(j)}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\mathcal{P}}_i^{(j)}, \quad \bar{\boldsymbol{\mathcal{P}}}=\frac{1}{M}\sum_{j=1}^{M}\bar{\boldsymbol{\mathcal{P}}}^{(j)}, \end{aligned} \tag{3.29}$$

$$\boldsymbol{W}=\frac{1}{M}\sum_{j=1}^{M}\boldsymbol{s}_j^2, \quad \text{where} \quad \boldsymbol{s}_j^2=\frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{\mathcal{P}}_i^{(j)}-\bar{\boldsymbol{\mathcal{P}}}^{(j)}\right)\left(\boldsymbol{\mathcal{P}}_i^{(j)}-\bar{\boldsymbol{\mathcal{P}}}^{(j)}\right)^{\intercal}. \tag{3.30}$$

MPSRF is then defined as

$$\widehat{R}^p=\frac{N-1}{N}+\frac{M+1}{M}\lambda_1, \tag{3.31}$$

where λ_1 is the largest eigenvalue of $\boldsymbol{W}^{-1}\boldsymbol{B}$. Similar to (3.28), $\widehat{R}^p$ approaches 1 from above and consensus is that $\widehat{R}^p < 1.1$ is an indication of convergence.

I compute both (3.28) and (3.31) and present the results in Table 3.8. For both mechanisms, analysis A finds all of these potential scale reduction factors to be greater than 1.1, indicating a lack of convergence. For the second and third analyses, all statistics are less than 1.1, indicating convergence. It is noteworthy that for analysis B the potential reduction is roughly 10^{-2} and for analysis C the reduction factor is 10^{-3}, so analysis C shows stronger evidence of convergence.

Investigating trace plots and computing $\widehat{R}$ are the primary means of assessing posterior convergence. However, samples in MCMC chains exhibit strong autocorrelation: correlation between a sample and the samples that come before and after it. This autocorrelation arises due to the Markov Chain that is constructed during MCMC algorithms. Therefore, consecutive samples provide similar information. Instead it would be preferable to have independent samples, particularly for the purposes of performing Monte Carlo integration over parameter space like I do in Chapter 4 and Chapter 5. Independent samples also provide a more intuitive understanding of how much information one has. Practical experience from [35] suggests that 10 to 100 independent samples in each chain is sufficient for many cases.

The way to find how many independent samples exist within one's chains is to compute the so-called *autocorrelation length scale* – the average distance between samples before they are independent – and divide the total number of samples by that. The number of independent samples is called the *effective sample size* and I denote it $\widehat{n}_{\text{eff}}$; the "hat" is because one has to estimate this quantity. From [35], the way to compute $\widehat{n}_{\text{eff}}$ is

$$V_t = \frac{1}{M(N-t)} \sum_{j=1}^{M} \sum_{i=t+1}^{N} \left(\mathcal{P}_i^{(j)} - \mathcal{P}_{i-t}^{(j)} \right)^2 \tag{3.32}$$

$$\widehat{\rho}_t = 1 - \frac{V_t}{2\widehat{\text{var}}^+} \tag{3.33}$$

$$\widehat{n}_{\text{eff}} = \frac{NM}{1 + 2\sum_{t=1}^{T} \widehat{\rho}_t}, \tag{3.34}$$

where M is the number of chains, N is the number of samples per chain, and T is a cutoff where the correlation $\widehat{\rho}_t$ is too noisy. In [35], the convention is to take T to be the first, odd positive integer for which $\widehat{\rho}_{T+1} + \widehat{\rho}_{T+1} < 0$.

I implement the computation of (3.34). Figure 3.21 visualizes $\widehat{\rho}_t$ versus the lag t for every parameter in the 3-step and 4-step mechanism. The lag t for which all of the curves collapse to zero is a visual cue for the number of samples that need to be skipped to find an independent sample. Table 3.9 shows the computation of the effective sample size for each MCMC analysis. For both mechanisms, I find that analysis A contained very few independent samples, finding between 1–65

Table 3.9: *Effective sample size for each parameter for each MCMC analysis as calculated by* (3.34)*. Effective sample size describes the number of independent samples and, since all of my analyses have the same number of chains and samples per chain, also acts as a comparison of the efficiency of the sampling performed.*

Analysis	k_f	k_b	k_1	k_2	k_3	k_4	M
3-Step A	44	104	31	38	125	–	1315
3-Step B	1018	1236	843	984	1115	–	1139
3-Step C	11876	11883	10887	10626	11944	–	11691
4-Step A	57	43	27	30	58	58	97
4-Step B	1656	1729	1494	1546	1443	1615	1645
4-Step C	11632	12057	9328	9007	10979	9403	12094

independent samples per chain. In analysis B, although the convergence statistics in Table 3.8 indicate analysis B converged, the low acceptance ratios caused substantial autocorrelation. However, there is still in improvement over analysis A with 42–86 independent samples per chain. Finally, in analysis C, the sampling was much more efficient and I produced 450–604 independent samples per chain.

I also look at the relationship between the acceptance ratio of an MCMC analysis and the effective sample size. Figure 3.20 shows box plots of the acceptance ratio for each chain in each MCMC analysis I conduct. Analysis A of both the 3-step and 4-step mechanism did not converge to the posterior, neither had many independent samples, and the 3-step analysis was near "optimal" acceptance ratios whereas the 4-step was far below. This result tells me that if one has non-convergent chains then one cannot expect many independent samples regardless of acceptance ratio. The second and third analyses both converged according to my analysis above, but the acceptance ratio of the third analyses were far closer to "optimal" than the second. I interpret this result as what "optimal" acceptance ratios really mean: one receives a more efficient gain of information per sample.

I have conducted a detailed analyses of the convergence of my posterior distributions. The results I computed in the third MCMC analyses are sufficient for me to have an accurate representation of the posterior distribution. With this knowledge, I have enough information to quantitatively select a model. The next section discusses this process.

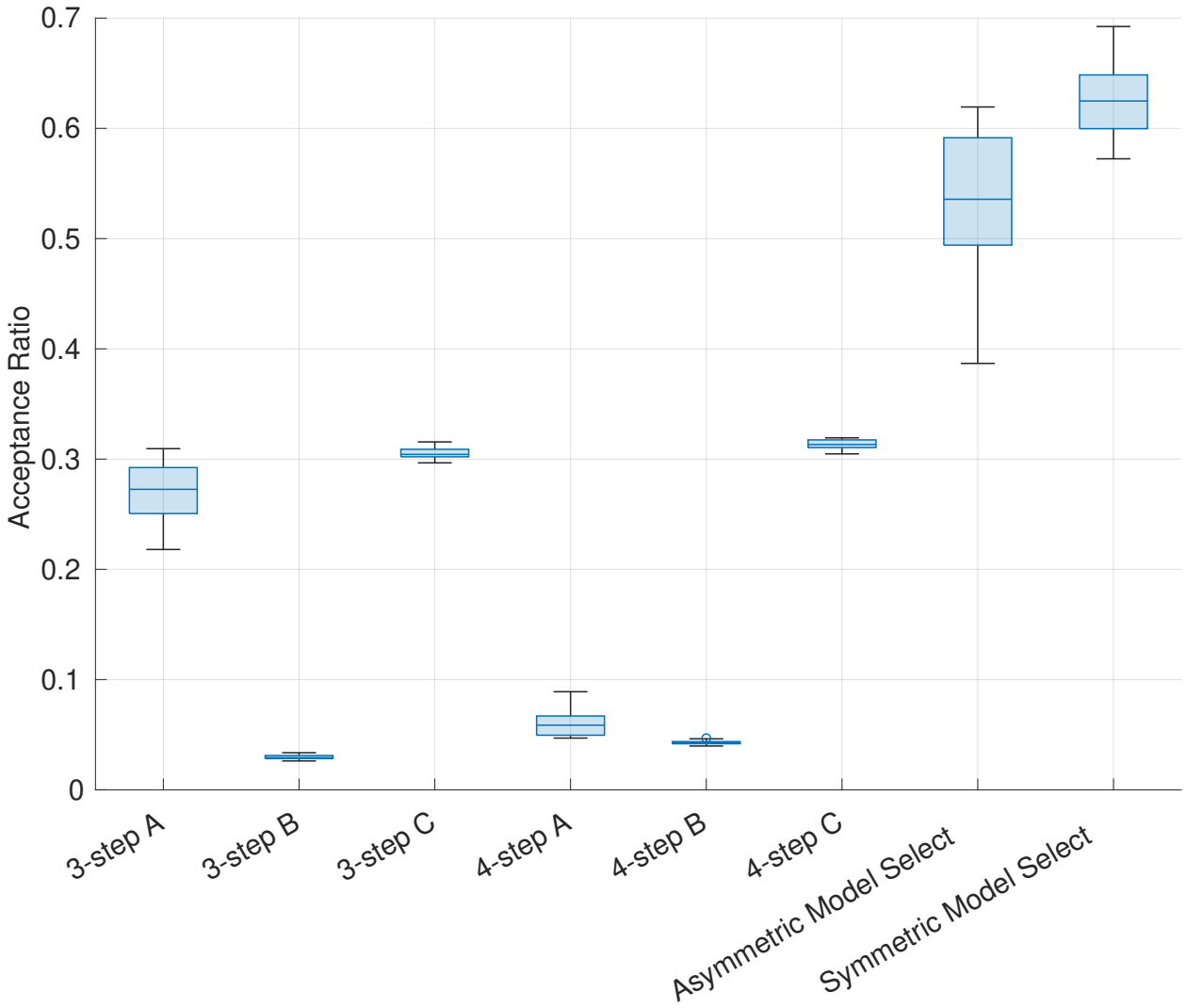


Figure 3.20: *Box plot showing the acceptance ratios of the chains in the analyses performed in Section 3.5 and Section 3.7. Each box displays the median value with a blue horizontal line, the* $2.5\% - 97.5\%$ *quantile outlined in blue, the minimum and maximum non-outlier values in black, and any outliers with blue circles.*

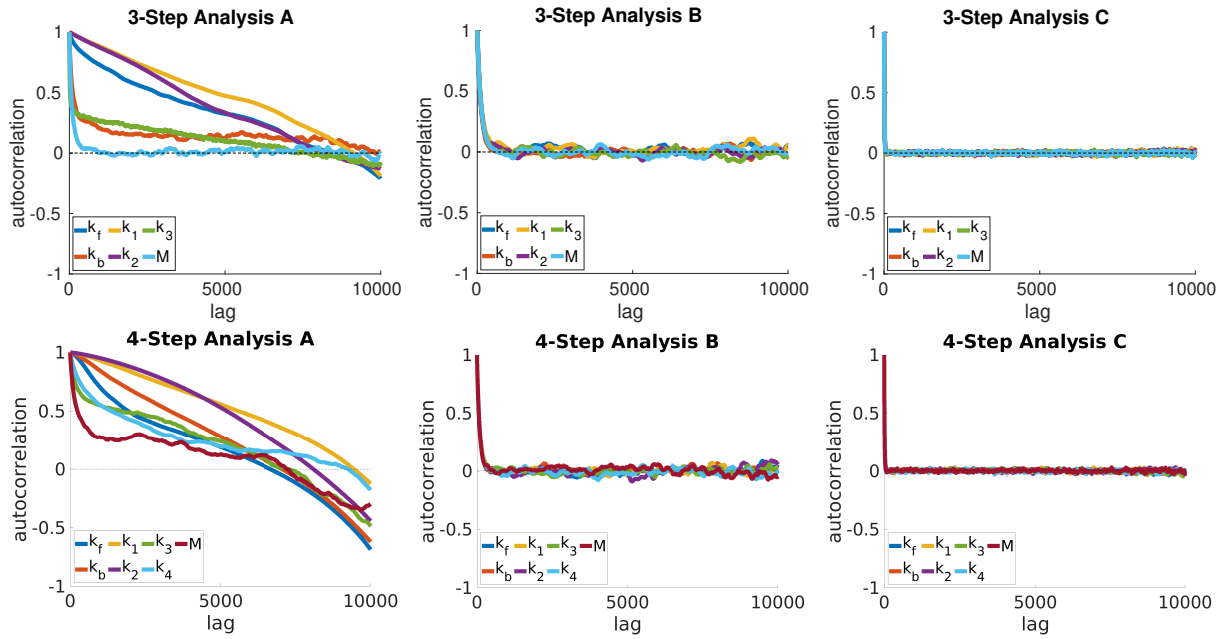


Figure 3.21: *Autocorrelation function estimates* (3.33) *for both MCMC analysis of the 3- and 4-step mechanisms. When thinning MCMC chains to produce chains with close to independent samples, every ℓth sample is taken where ℓ is the lag for which all parameter autocorrelations have collapsed to zero.*

3.7 Model selection

This chapter, so far, has described the process of generating samples such that the distribution of the parameters mimics, and eventually converges to, the posterior distribution. In the context of scientific development, one often wants to repeat the process determining the posterior for multiple different models. Once samples are generated for multiple models, the natural next step is to quantitatively determine which model best describes the data. This quantitative assessment is done by computing the *Bayes factor* [56] between two or more competing models. I find that the 3-step mechanism is the more appropriate model. This was also the determination in [48, 49], but my application of Bayesian inversion allows me to make this decision in a more rigorous manner.

In this section, I will derive the equations for the Bayes factor, describe the heuristic interpretation of the Bayes factor, and provide the computational means of computing the Bayes factor provided sampled posterior distributions. As I will discuss, there are significant difficulties with assessing model selection by use of Bayes factors. Hence, I will also describe an alternative method that poses a mixture model formulation blending the two models together and allow MCMC to

elucidate the preferred model. I apply these methods to comparing the 3-step and 4-step mechanisms and walk through my interpretation of the results that led me to determine that the 3-step mechanism is the appropriate model.

3.7.1 Bayes' factor

Let $\mathcal{M}$ denote a computational model, for example the 3-step and 4-step mechanisms are different computational models. From Bayes' theorem, I can express the posterior distribution of a computational model $\pi(\mathcal{M}|\text{data})$ in a similar manner to (3.1) with

$$\pi(\mathcal{M}|\text{data}) = \frac{\pi(\text{data}|\mathcal{M})\pi(\mathcal{M})}{\pi(\text{data})}. \tag{3.35}$$

I am interested in $\pi(\text{data}|\mathcal{M})$ as this represents the probability of finding the experimental data assuming $\mathcal{M}$ is the true model; in other words, $\pi(\text{data}|\mathcal{M})$ represents the support $\mathcal{M}$ provides for collecting the observed data. Therefore, the philosophy of data-driven model selection insists that the model for which the data is best supported is the appropriate choice of model. Bayes factor is a comparison between two models, $\mathcal{M}_1$ and $\mathcal{M}_2$, given by

$$\text{BF}_{12} = \frac{\pi(\text{data}|\mathcal{M}_1)}{\pi(\text{data}|\mathcal{M}_2)}, \tag{3.36}$$

so that $\text{BF}_{12} > 1$ indicates support for $\mathcal{M}_1$, $\text{BF}_{12} < 1$ indicates support for $\mathcal{M}_2$, and $\text{BF}_{12} = 1$ indicates $\mathcal{M}_1$ and $\mathcal{M}_2$ are equally likely.

As always, data is noisy and $\text{BF} > 1$ could simply be an artifact of a combination of measurement error and sample size bias. Hence, heuristic scales have been developed to interpret the Bayes factor. I use a popular table from [60] reproduced in Table 3.10 for my heuristic interpretation of Bayes factor.

Table 3.10: *Interpretation of Bayes factor values as a measure of the strength of evidence. This table is reproduced from [60].*

$\log_{10}$ BF	BF	Strength of evidence
0 to 1/2	1 to 3.2	Not worth more than a bare mention
1/2 to 1	3.2 to 10	Substantial
1 to 2	10 to 100	Strong
>2	>100	Decisive

Since the models, $\mathcal{M}_i$, I am considering depend on parameters $\mathcal{K}_i$, $\pi(\text{data}|\mathcal{M}_i)$ is obtained by integrating over parameter space. Hence,

$$\pi(\text{data}|\mathcal{M}_i) = \int_{D(\mathcal{K}_i)} \pi_{\text{L}}(\text{data}|\mathcal{K}_i, \mathcal{M}_i)\pi_{\text{pr}}(\mathcal{K}_i|\mathcal{M}_i)d\mathcal{K}_i, \tag{3.37}$$

where $D(\mathcal{K}_i)$ is the parameter space for $\mathcal{M}_i$. (3.37) can be made explicit by indicating the dependence on $\mathcal{M}$ in (3.1) and noticing

$$\begin{aligned} \pi(\mathcal{K}|\text{data}, \mathcal{M}) &= \frac{\pi_{\text{L}}(\text{data}|\mathcal{K}, \mathcal{M})\pi_{\text{pr}}(\mathcal{K}|\mathcal{M})}{\pi(\text{data}|\mathcal{M})}, \\ \int_{D(\mathcal{K})} \pi(\mathcal{K}|\text{data}, \mathcal{M})d\mathcal{K} &= \int_{D(\mathcal{K})} \frac{\pi_{\text{L}}(\text{data}|\mathcal{K}, \mathcal{M})\pi_{\text{pr}}(\mathcal{K}|\mathcal{M})}{\pi(\text{data}|\mathcal{M})}d\mathcal{K}, \\ 1 &= \frac{1}{\pi(\text{data}|\mathcal{M})}\int_{D(\mathcal{K})} \pi_{\text{L}}(\text{data}|\mathcal{K}, \mathcal{M})\pi_{\text{pr}}(\mathcal{K}|\mathcal{M})d\mathcal{K}, \\ \pi(\text{data}|\mathcal{M}) &= \int_{D(\mathcal{K})} \pi_{\text{L}}(\text{data}|\mathcal{K}, \mathcal{M})\pi_{\text{pr}}(\mathcal{K}|\mathcal{M})d\mathcal{K}. \end{aligned} \tag{3.38}$$

Note that $\pi(\text{data}|\mathcal{M})$ is simply the normalization factor for $\pi_{\text{L}}(\text{data}|\mathcal{K}, \mathcal{M})\pi_{\text{pr}}(\mathcal{K}|\mathcal{M})$. Since I cannot analytically compute the integral in (3.37), I must use estimators based on the samples generated using the techniques of Section 3.4. In the remainder of my thesis, I will denote an estimator of $\pi(\text{data}|\mathcal{M})$ as $\widehat{\pi}(\text{data})$ and drop the dependence on $\mathcal{M}$ except when I need to compare results from multiple models.

To make the most out of MCMC samples, the most commonly used method to estimate (3.37) is called the harmonic mean estimator (HME) [78] given by

$$\widehat{\pi}_{\text{HME}} = \left[\frac{1}{N} \sum_{n=1}^{N} \frac{1}{\pi_L \left(\text{data} | \mathcal{K}^{(n)} \right)} \right]^{-1}. \tag{3.39}$$

Applying (3.39) to the 3-step and 4-step mechanism, I find

$$\log_{10} \left(BF_{34} \right) = \log_{10} \left(\frac{\pi(\text{data}|\text{3-step})}{\pi(\text{data}|\text{4-step})} \right) \approx \log_{10} \left(\frac{\widehat{\pi}_{\text{HME}}(\text{3-step})}{\widehat{\pi}_{\text{HME}}(\text{4-step})} \right) = 5.62, \tag{3.40}$$

indicating "decisive" evidence for the 3-step mechanism. However, I must note that the HME estimator is flawed. In general, the convergence rate of (3.39) is $N^{1/\alpha - 1}$ where $1 < \alpha \leq 2$ and $\alpha \approx 1$ in cases where the prior is much broader than the likelihood [19]. Alternative methods have been proposed, e.g. see [19, 30] for reviews, but they require relatively involved calculations. The appeal of (3.39) is that the calculation is a simple post-processing step, so it would be nice to have a method that is as simple as that. That desire leads to an alternative methodology which I describe next.

3.7.2 Mixture model

Recently, a new approach to the model selection problem was proposed in [89]. This proposed replacement considers a mixture of two computational models – or, in theory, more than two, but I will stick with two for the purposes of my thesis – each with their own set of parameters, and the extent to which the models are mixed is also controlled by a (hyper-)parameter. More formally, consider the models $\mathcal{M}_1$ and $\mathcal{M}_2$ with their corresponding parameters $\mathcal{K}_1$ and $\mathcal{K}_2$. Also consider the mixture parameter $\omega \in [0, 1]$. Then, let

$$\mathcal{M}(\mathcal{K}, \omega) = \omega \mathcal{M}_1(\mathcal{K}_1) + (1 - \omega) \mathcal{M}_2(\mathcal{K}_2), \tag{3.41}$$

where $\mathcal{K} = \{\mathcal{K}_1, \mathcal{K}_2\}$ and addition represents adding together the resulting quantities from the computational models. In my case, each computational model is the solution of the ODEs to obtain particle concentrations and thus addition is defined as adding together the concentrations for each particle. The mixture model $\mathcal{M}(\mathcal{K}, \omega)$ can then be treated as a Bayesian inference problem where the result of ω indicates which model is preferred. Depending on the relationship between $\mathcal{M}_1$ and $\mathcal{M}_2$, a prior distribution can be assigned to ω to reflect preferences in the model selection process. For instance, if the two models have similar levels of complexity, then a prior distribution that is symmetric and has most of its mass near zero and one will impart a desire to use one model or the other – but not both! – and assign equal prior probability to each model. An example of how to accomplish this is with the beta distribution, whose probability distribution function (PDF) is

$$\begin{aligned} \mathcal{B}(\omega; \alpha, \beta) &= \frac{\omega^{\alpha-1}(1-\omega)^{\beta-1}}{B(\alpha, \beta)}, \\ B(\alpha, \beta) &= \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \end{aligned} \tag{3.42}$$

where Γ is the Gamma function. When the shape parameters α, β of the beta distribution are equal and both less than one, the PDF is u-shaped as depicted in the left plot of Figure 3.22. On the other hand, if the shape parameters are less than one, but unequal, asymmetry is introduced as shown in the right plot of Figure 3.22. This prior distribution is suitable for models with varying levels of complexity where the more simple model is preferred. I test both of the cases shown in Figure 3.22 to see if the results vary depending on whether the 3-step is deemed equally or more likely.

For my specific problem of model selection, let $\mathcal{M}_{\text{3-step}}(\mathcal{K}_{\text{3-step}})$ and $\mathcal{M}_{\text{4-step}}(\mathcal{K}_{\text{4-step}})$ denote the computational models for the 3-step and 4-step mechanisms respectively. Then I form the mixture model

$$\mathcal{M}(\mathcal{K}_{\text{3-step}}, \mathcal{K}_{\text{4-step}}, \omega) = \omega\mathcal{M}_{\text{3-step}}(\mathcal{K}_{\text{3-step}}) + (1-\omega)\mathcal{M}_{\text{4-step}}(\mathcal{K}_{\text{4-step}}). \tag{3.43}$$

Previously, the computation of the likelihood involved solving the ODEs for $\mathcal{M}_{\text{3-step}}(\mathcal{K}_{\text{3-step}})$ or $\mathcal{M}_{\text{4-step}}(\mathcal{K}_{\text{4-step}})$, translating the concentrations into probabilities with (3.3) and (3.5), and then com-

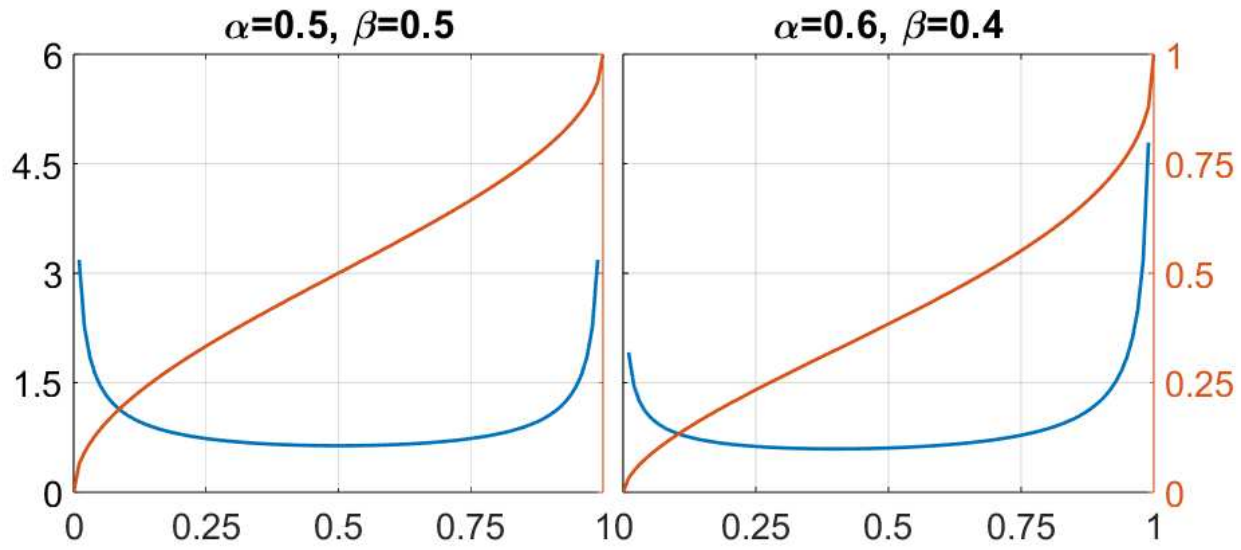


Figure 3.22: *Prior distributions for the mixture parameter ω in* (3.41). *(Left) A prior distribution suitable for equally favoring both models. Note the cumulative distribution function (CDF) in orange reaches* 0.5 *at* $\omega = 0.5$ *indicating an equal weighting towards either model. (Right) A prior distribution suitable for favoring one model over the other. Note the cumulative distribution function (CDF) in orange reaches* 0.5 *when* $\omega > 0.5$ *indicating weighting towards the 3-step mechanism.*

puting the resulting multinomial distribution with the observed data as input. Now, the model is a weighted average between the 3-step and 4-step mechanisms, so once the concentrations are computed for both mechanisms, they are weighted together with ω to compute the mixture model concentrations. Then the likelihood calculations proceeds in the same manner as I described in Section 3.2. I can then run my same MCMC procedure, but this time using the mixture model. Here, I take as my prior distribution

$$\begin{aligned} \pi_{\text{pr}}\left(\mathcal{K}_{\text{3-step}}\right) &\sim N\left(\mathcal{K}^*_{\text{3-step}}, C_{\text{3-step}}\right) \\ \pi_{\text{pr}}\left(\mathcal{K}_{\text{4-step}}\right) &\sim N\left(\mathcal{K}^*_{\text{4-step}}, C_{\text{4-step}}\right) \\ \omega &\sim \mathcal{B}(0.5, 0.5) \quad \text{or} \quad \mathcal{B}(0.6, 0.4), \end{aligned} \tag{3.44}$$

where $\mathcal{K}^*_{\text{3-step}}$ and $\mathcal{K}^*_{\text{4-step}}$ are the *maximum a posteriori* parameters noted in Table 3.4 and Table 3.7, $C_{\text{3-step}}$ and $C_{\text{4-step}}$ are the covariance matrices of the posterior samples from the third MCMC analysis performed in Section 3.5, and there is no correlation between each of the three, disjoint pa-

rameter sets. I take the proposal distribution to be

$$\begin{bmatrix} \mathcal{K}_{\text{3-step}} \\ \mathcal{K}_{\text{4-step}} \\ \omega \end{bmatrix}^{\text{trial}} \sim N\left(\begin{bmatrix} \mathcal{K}_{\text{3-step}} \\ \mathcal{K}_{\text{4-step}} \\ \omega \end{bmatrix}^{\text{previous}}, \begin{pmatrix} C_{\text{3-step}} & & \\ & C_{\text{4-step}} & \\ & & 1 \end{pmatrix} \right). \tag{3.45}$$

I manually tune this proposal distribution until I find the first 100 samples on each of the 20 chains has an acceptance ratio within 20%–50%. In this case, $\mathcal{K}_{\text{3-step}}$ and $\mathcal{K}_{\text{4-step}}$ are *nuisance* parameters that I am not interested in, so my decisions are based more in line with what is optimal for a one-dimensional MCMC problem: a higher acceptance ratio of 44% is theoretically optimal in special cases [37]. I then run 20 chains of 5000 samples using the Metropolis-Hastings algorithm. I denote the analysis where $\omega \sim \mathcal{B}(0.5, 0.5)$ as the "unbiased" analysis (unbiased, here, used as a layperson's term rather than the formal statistical definition) and the analysis where $\omega \sim \mathcal{B}(0.6, 0.4)$ as the "biased" analysis.

After running an MCMC analysis on the mixture model, I integrate over the *nuisance* parameters $\mathcal{K}_{\text{3-step}}$ and $\mathcal{K}_{\text{4-step}}$; in practice, this means simply ignoring all parameter values other than ω and performing the rest of the analysis as if ω was the only parameter. I test convergence as described in Section 3.6. I find that

$$\widehat{R}_{\text{unbiased}} = 1.019 \tag{3.46}$$

$$\widehat{R}_{\text{biased}} = 1.024, \tag{3.47}$$

indicating convergence, and the trace plots in Figure 3.23 also indicate convergence to the posterior, with a burn-in period of approximately 200 samples. Graphing the posterior distribution for ω clearly shows where the evidence lies. Figure 3.24 shows the posterior for both the biased and unbiased analyses. These figures plot both the PDF and cumulative distribution function (CDF) of the posterior. The important quantity to note is what value of ω corresponds to the CDF being 0.5; call that point ω^*. If $\omega^* < 0.5$, then the evidence suggests the 4-step mechanism is preferable. If

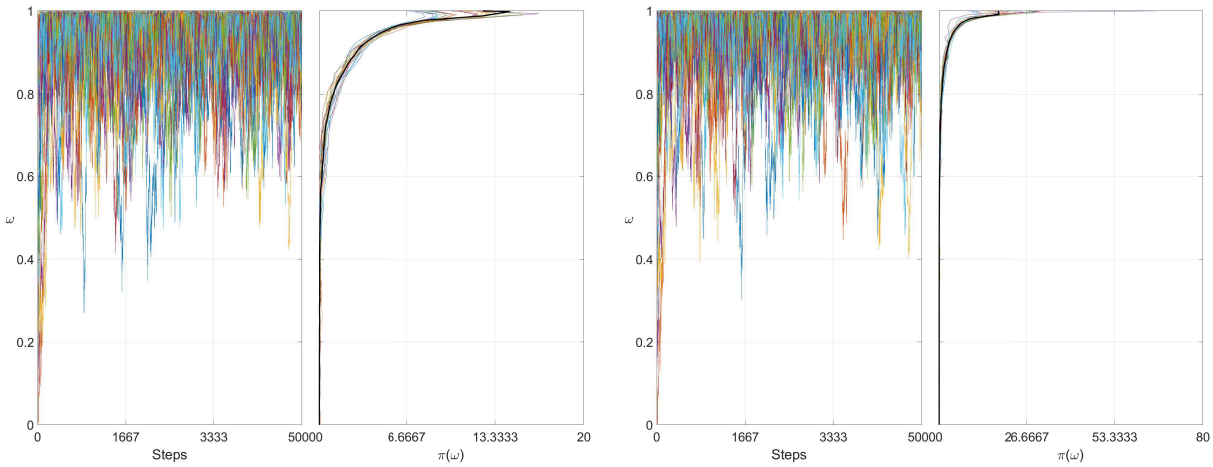


Figure 3.23: *Trace plots for the model selection MCMC analysis. (left) The results using an unbiased prior on ω. (right) The results using a prior distribution on ω biased towards the 3-step mechanism.*

$\omega^* > 0.5$, then the 3-step mechanism is preferred. Moreover, how close ω^* is to 0.5 changes how definitive the evidence is. In both cases, the evidence strongly favors the 3-step mechanism. I find $\omega^*_{\text{unbiased}} = 0.9567$ and $\omega^*_{\text{biased}} = 0.9759$.

My analysis throughout this entire chapter provided three different ways to select between the 3-step and 4-step mechanisms. First, in Section 3.5.3, the principle of Occam's razor would suggest that since the 3-step mechanism is more simple than the 4-step and the fits to the data are qualitatively similar, that the 3-step mechanism is the appropriate choice: make a model precisely as complicated as it needs to be. Second, in Section 3.7.1, the (highly flawed, but simple) HME computation of the Bayes factor suggested strong evidence for the 3-step mechanism. Third, this section demonstrated with the mixture model approach that the 3-step mechanism is the model supported best by the data. Therefore I conclude that the 3-step is the *minimal* mechanism necessary to describe this nanoparticle system.

The results of this chapter provide an important scientific advancement: the detailed characterization of not only the appropriate mathematical model to use but also the parameter uncertainty present. Recall, however, the ultimate goal of studying nanoparticle synthesis that I described in Chapter 1. I want to be able to manipulate experimental conditions so that a chemical reaction results in a near-monodisperse PSD. Then, since nanoparticles have size-dependent properties, the

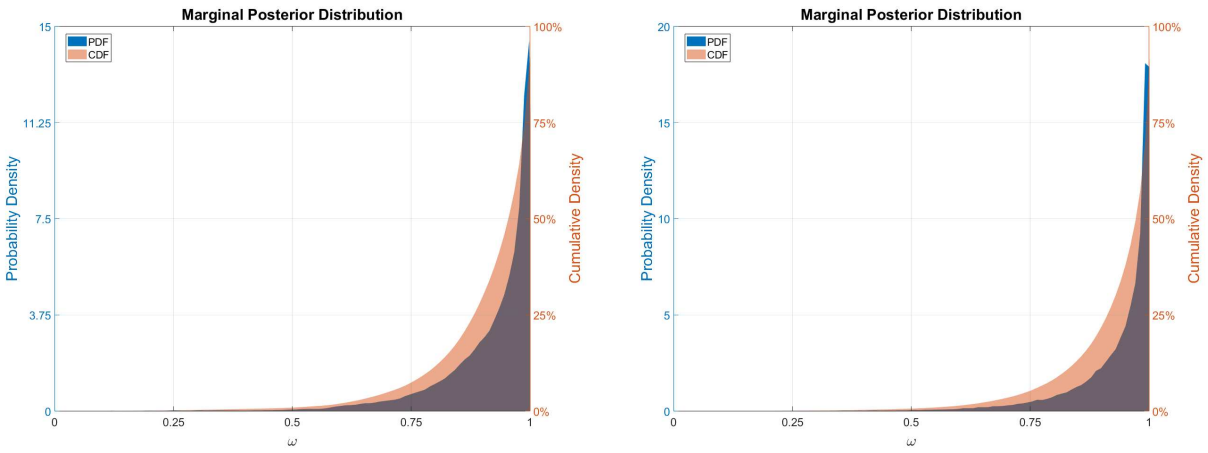


Figure 3.24: *Posterior distribution plotted for ω in the model selection MCMC analysis. Each plot contains the PDF and CDF of the posterior distribution. For model selection purposes, the value ω^* where the CDF reaches 50% indicates which model was selected. Here, $\omega^* > 0.5$ indicates evidence for the 3-step mechanism and $\omega^* < 0.5$ indicates evidence for the 4-step mechanism.*

outcome of the reaction will have properties more specific to one's needs. With the Bayesian inverse problem I performed in this chapter, I am well-equipped to perform simulations of particle concentrations over time and quantify the uncertainty present in the particle concentrations. Because I can simulate the chemical reaction, I can use these simulations in the setting of an optimization problem to adjust the experiment closer to what I desire. This optimization problem is the subject of the next chapter.

Chapter 4

Optimization under uncertainty

A key question in nanoparticle synthesis is how to create more narrow particle size distribution (PSD)s in order to produce more specific properties from the resultant set of nanoparticles. With the completion of the Bayesian inversion analysis of Chapter 3, I have selected the 3-step mechanism and I am able to simulate a reaction along with the uncertainty around the particle concentrations. Now I want to explore how modifying experimental conditions affects the PSD and develop a method to find experimental conditions that lead to a desired outcome. In this chapter, I focus on the desired outcome being a near-monodisperse PSD centered around some prechosen particle size.

The optimization performed in this chapter will involve new parameters. To incorporate uncertainty within the optimization I perform, I will need to perform Monte Carlo integration over the model parameters. That is, $\mathcal{K}_{\text{3-step}}$ will refer to the model parameters discussed in Chapter 3 and will follow the posterior $\pi(\mathcal{K}_{\text{3-step}}|\text{data})$ visualized in Figure 3.5 that I constructed during the final MCMC analysis. For brevity I will drop the "3-step" and simply refer to these parameters as $\mathcal{K}$. I introduce new parameters θ which denote the parameters I optimize over. These optimization parameters include the end time of the reaction, the initial precursor concentration, and the initial ligand concentration as experimental conditions that are easy to manipulate. I also include modifications to model parameters k_f, k_1, k_2, and k_3 as an exploratory "what if" analysis. These model parameters *can* be effectively modified by methods such as adjusting the temperature, using a different ligand, using a different solvent concentration, or using a different solvent altogether. However, the precise relationship between, e.g., a change in temperature of $10\,^{\circ}\mathrm{C}$ and the corresponding change in k_1, k_2, and k_3 is currently unknown and requires further research. Despite this limitation, hypothetical studies are valuable in understanding if, for example, studying temperature dependence could lead to narrower PSDs.

In this chapter I will begin in Section 4.1 by defining the optimization problem I want to solve. Then I will discuss how I evaluate the cost function that quantifies how well a particular θ fits my goals in Section 4.2. Then, in Section 4.3 I discuss the optimization algorithm I use and finally in Section 4.4 I will present my results.

4.1 Mathematical formulation

The first goal I have is to identify the quantities that matter. As I previously described, nanoparticles have size-dependent properties, so I would like to control the reactions to get as many particles as possible near the desired size. As such, considering the mean value and variance of the PSD are crucial. Hence, I care about the two functions

$$PM(\theta;\mathcal{K}) = \frac{\sum_{i=3}^{2500} B_i(\theta;\mathcal{K})\text{diameter}(i)}{\sum_{i=3}^{2500} B_i(\theta;\mathcal{K})}, \tag{4.1}$$

$$PS(\theta;\mathcal{K}) = \sqrt{\frac{\sum_{i=3}^{2500} B_i(\theta;\mathcal{K})\left[\text{diameter}(i) - PM(\theta;\mathcal{K})\right]^2}{\sum_{i=3}^{2500} B_i(\theta;\mathcal{K})}}, \tag{4.2}$$

where $B_i(\theta;\mathcal{K})$ is the concentration of particles with i atoms from solving the ODEs (2.13), diameter(i) is the conversion from number of atoms to the diameter in nanometers given by (3.4), PM stands for "PSD mean", and PS "PSD standard deviation". However, with respect to the mean of the PSD, I care about how close it is to a given desired size, which I denote μ, giving me the function

$$MD(\theta;\mathcal{K}) = \left|PM(\theta;\mathcal{K}) - \mu\right|, \tag{4.3}$$

where MD stands for "mean deviation".

Given the uncertainty captured in the Bayesian inference described in Chapter 3, I can account for the distributions in the inferred parameters. Thus, I can take expectations over the space of

inferred parameters $\mathcal{K}$, with expectation of some function f defined as

$$\mathbb{E}(f) = \int f(\mathcal{K})\pi(\mathcal{K}|\text{data})d\mathcal{K}, \tag{4.4}$$

where $\pi(\mathcal{K}|\text{data})$ is the posterior distribution computed in Chapter 3, giving functions

$$EMD(\theta) = \mathbb{E}\left(MD(\theta;\mathcal{K})\right), \tag{4.5}$$

$$EPS(\theta) = \mathbb{E}\left(PS(\theta;\mathcal{K})\right). \tag{4.6}$$

Now, from a practical point of view, one wants a consistent yield from their nanoparticle synthesis process. In other words, the distribution about the inferred parameters induces a distribution about the mean and variance of the PSD. As a result, it may be the case that on average the mean of the PSD is close to the desired mean, but the variance of the PSD mean is quite large, and similarly for the variance of the PSD. Thus, incorporating the variability due to the inferred parameters in my optimization could be useful. Therefore, I also consider the functions

$$VMD(\theta) = \mathbb{E}\left(\left[MD(\theta;\mathcal{K}) - EMD(\theta)\right)^2\right], \tag{4.7}$$

$$VPS(\theta) = \mathbb{E}\left(\left[PS(\theta;\mathcal{K}) - EPS(\theta)\right]^2\right). \tag{4.8}$$

Finally, it would be wasteful for the reaction to not run to completion. Therefore, I also consider the conversion ratio

$$AC(\theta;\mathcal{K}) = \frac{A_{\text{end}}(\theta;\mathcal{K})}{A_0(\theta;\mathcal{K})}, \tag{4.9}$$

$$EAC(\theta) = \mathbb{E}(AC(\theta;\mathcal{K})), \tag{4.10}$$

where $A_{\text{end}}(\theta)$ is the precursor concentration when the reaction is stopped and $A_0(\theta)$ is the initial precursor concentration.

I am then left with the total cost function

$$\mathcal{C}(\theta) = w_1 EMD(\theta) + w_2 EPS(\theta) + w_3 VMD(\theta) + w_4 VPS(\theta) + w_5 EAC(\theta), \tag{4.11}$$

where w_i are weights given to each C_i. In my work I heuristically choose values for w_i and explain my choices in Section 4.4. This is called *scalarization* in the multi-objective optimization literature, but one could also approach this problem through alternative methods; see [41] for a recent review including different ways of choosing weights in the scalarization method.

Now, I have a cost function that is the weighted sum of five expectations; that is, the weighted sum of five difficult integrals which I cannot compute exactly. In the following section I will describe how I will construct an approximation to the cost function that is feasible to numerically compute.

4.2 Evaluation of the cost function

Evaluating the cost function (4.11) involves computing expectations, which is essentially propagating the uncertainty in $\mathcal{K}$ through the ODEs (2.13) to compute a distribution for MD, PS, and AC, and finally computing summary statistics in the form of mean and variance. The simplest way to perform this process is through Monte Carlo integration, i.e. for $\left\{\mathcal{K}^{(i)}\right\}_{i=1,\dots,N}$ drawn from $\pi(\mathcal{K}|\text{data})$

$$\begin{aligned}
\widehat{EMD}(\theta) &= \frac{1}{N}\sum_{i=1}^{N} MD(\theta;\mathcal{K}^{(i)}), \quad & \widehat{VMD}(\theta) &= \frac{1}{N}\sum_{i=1}^{N}\left[MD(\theta;\mathcal{K}^{(i)}) - \widehat{EMD}(\theta)\right]^2, \\
\widehat{EPS}(\theta) &= \frac{1}{N}\sum_{i=1}^{N} PS(\theta;\mathcal{K}^{(i)}), & \widehat{VPS}(\theta) &= \frac{1}{N}\sum_{i=1}^{N}\left[PS(\theta;\mathcal{K}^{(i)}) - \widehat{EPS}(\theta)\right]^2, \\
\widehat{EAC}(\theta) &= \frac{1}{N}\sum_{i=1}^{N} AC(\theta;\mathcal{K}^{(i)}). &&
\end{aligned} \tag{4.12}$$

Monte Carlo integration is the technique I employ for my work since I found the computational expense of computing the approximations (4.12) to be acceptable; see Table 4.1. Monte Carlo

integration remains the most computationally efficient integration strategy for high dimensional integrals (approximately > 10 in this context), but polynomial chaos methods could have also been used in this context to calculate mean values and variances with potentially fewer solves of the system of ODEs [80, 87, 120].

The question, then, is how many Monte Carlo samples are necessary to compute (4.12) to a reasonable accuracy. To test convergence, I choose θ corresponding to the experimental conditions of the data used in Chapter 3. I also choose a "desired size" of $2.7264\,\mathrm{nm}$ which corresponds to 750 atoms, which is close to the mean particle size at the end of the reaction under the same experimental conditions as the data in Chapter 3. I then draw $100,000$ values of $\mathcal{K}$ from $\pi(\mathcal{K}|\text{data})$, solve the ODEs for each $\mathcal{K}$, and record values for (4.2), (4.3), and (4.9). I then compute the values of the expectations in (4.12) as a function of the number of samples included in the integral. In Figures 4.1–4.5 I plot these results along with highlighting what values are within 5% and 1% of the final value. In Table 4.1 I note how many samples are necessary to be within 5% and 1% of the final value for all samples afterwards.

I encounter a complication, however, when I perform this optimization. As I discuss in Section 4.4, I introduce an initial concentration of POM as an optimization parameter. Physically, adding POM slows the reaction down and results in lower concentrations of particles. Pragmatically, this means I have to decrease the tolerance when I solve my ODEs which results in smaller time steps and thus a more computationally expensive solve. Solving 1000 ODEs each iteration of an optimization algorithm, then, is too computationally expensive for me. Therefore I decide to evaluate (4.12) using 500 samples and treat the evaluation of the cost function as a noisy – as opposed to exact – evaluation, which I explain further in Section 4.3.1.

Now that I have a methodology to evaluate my cost function, I discuss the algorithm I utilize to find the parameters θ that minimize my cost.

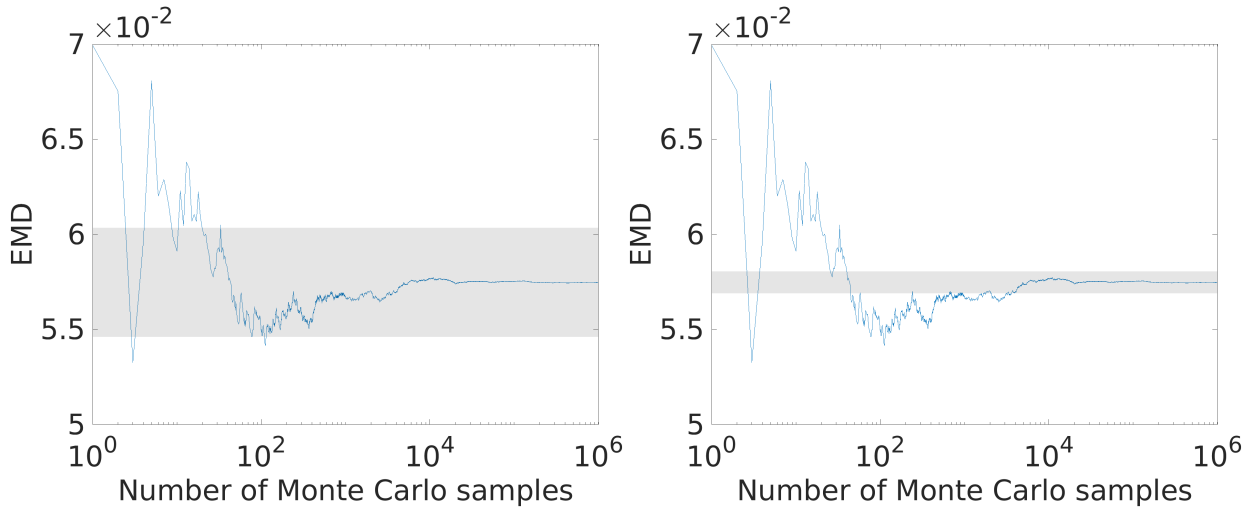


Figure 4.1: *Convergence of the Monte Carlo integration of* (4.5)*. The left plot shows* 5% *error bounds in gray. The right plot shows* 1% *error bounds in gray.*

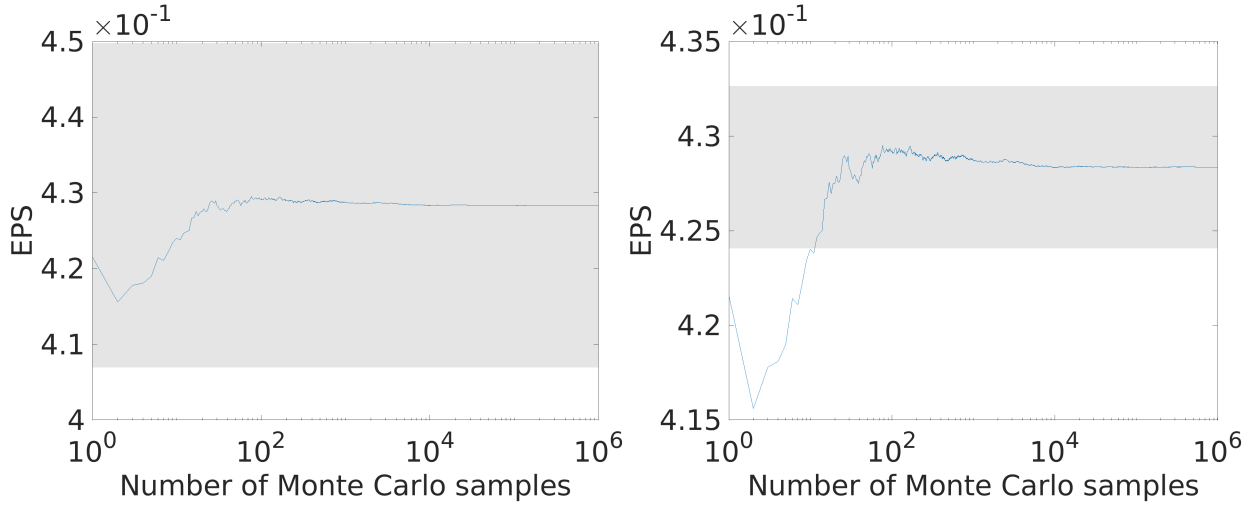


Figure 4.2: *Convergence of the Monte Carlo integration of* (4.6)*. The left plot shows* 5% *error bounds in gray. The right plot shows* 1% *error bounds in gray.*

Table 4.1: *Number of samples to reach a tolerance in Monte Carlo integration of the components of the cost function* (4.11)*. Tolerance is measured as the interval* $[(1-\varepsilon)x, (1+\varepsilon)x]$ *where* ε *is the tolerance level and* x *is the function value with the most Monte Carlo samples used.*

	Samples to be within	
Function	$\pm 5\%$	$\pm 1\%$
EMD	114	3,601
EPS	1	12
VMD	587	60,056
VPS	1,337	17,197
EAC	19	1,454

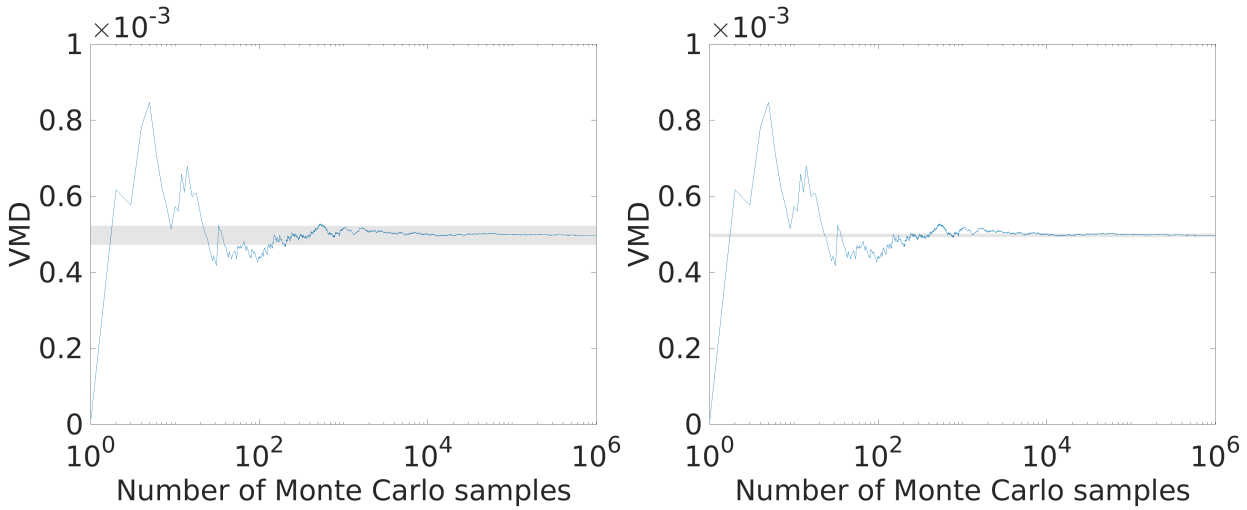


Figure 4.3: *Convergence of the Monte Carlo integration of* (4.7)*. The left plot shows* 5% *error bounds in gray. The right plot shows* 1% *error bounds in gray.*

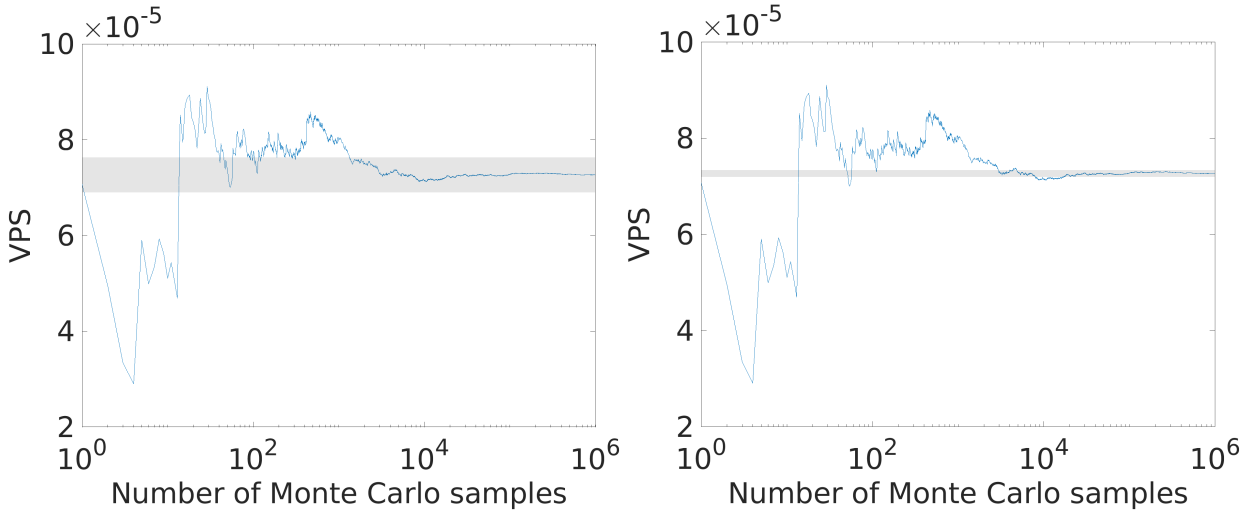


Figure 4.4: *Convergence of the Monte Carlo integration of* (4.8)*. The left plot shows* 5% *error bounds in gray. The right plot shows* 1% *error bounds in gray.*

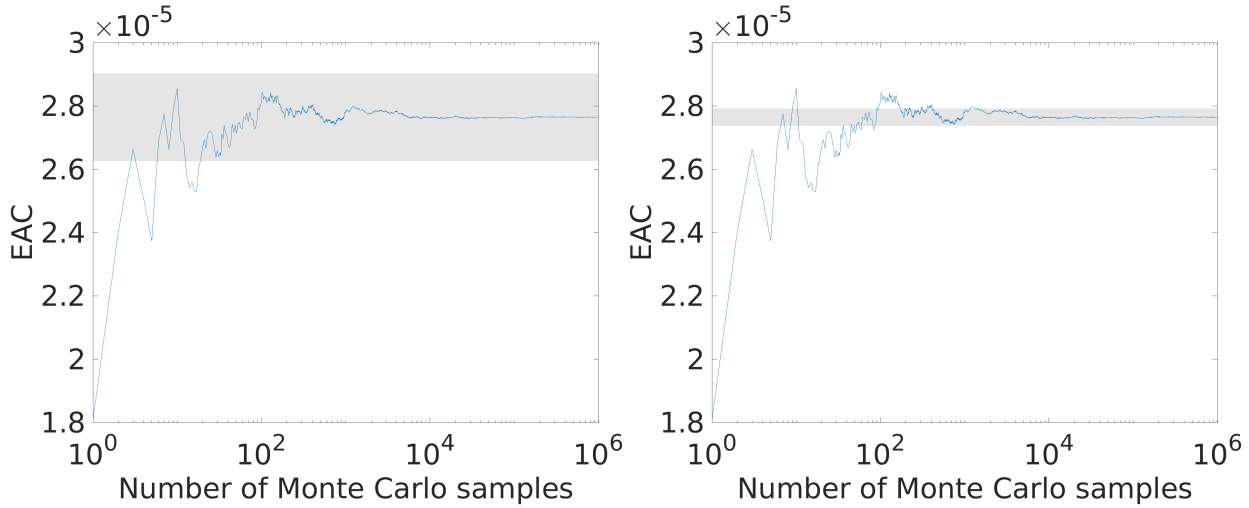


Figure 4.5: *Convergence of the Monte Carlo integration* (4.10). *The left plot shows* 5% *error bounds in gray. The right plot shows* 1% *error bounds in gray.*

4.3 Optimization algorithm

From Section 4.1 I have a cost function that I would like to minimize. However, I have two difficulties associated with optimization that must be addressed: (1) I cannot compute derivatives of $\mathcal{C}(\theta)$ and, as such, I cannot utilize algorithms like gradient descent, Newton's methods, and variants thereof; and (2) my cost function is the weighted sum of five different cost functions, which presumably each have their own, potentially non-overlapping, local and global minima, and hence it is reasonable to suspect that $\mathcal{C}(\theta)$ has many local minima. In this section I will describe the optimization algorithm I will use to remedy these two issues.

A number of methods have been developed to circumvent the lack of gradient information and yet still perform better than random search. A ubiquitous example is the Nelder-Mead algorithm [77]. In this algorithm, a simplex is constructed by evaluating the cost function at a number of points. The worst point – that is, the point corresponding to the largest evaluation of the cost function – is then reflected about the centroid of the opposite face of the simplex. If this reflected point is an improvement, then a new simplex is constructed and this process is repeated; if the reflected point is not an improvement, then some logic is applied to reduce the search area. Implementations vary for the exact algorithm, but the general idea is to follow a direction of decreasing cost function, which presumably is the direction that a gradient-based algorithm would also fol-

low. There are modern improvements [65, see Section 4.1 and references therein], but these are all from the class of derivative-free *local* optimization algorithms. All local algorithms are highly dependent on the starting point since they can only provide convergence to a *local* minimizer. In my problem, I cannot explicitly plot the cost function, so I cannot directly evaluate if I have a single, global minimizer or multiple local minimizers. Therefore, if I employed an algorithm such as Nelder-Mead, then it would be difficult to have confidence that my result is the global minimizer. From a practical perspective, this may mean that there are numerous experimental setups that are "pretty good", and perhaps that is good enough for certain practitioners, but it would be nice to find the best possible experimental setup.

In contrast, one can also pose so-called derivative-free *global* optimization algorithms. In principle, one might simply run Nelder-Mead many times starting at different points in the optimization domain. However, many algorithms have been posed to perform a global search in a more efficient manner, see [54, 61, 112] to name a few. In some sense, many of the common global optimization algorithms utilize clever brute-force. For example, in particle swarm optimization [10, 61], many "particles" are generated by choosing points within the optimization domain. Then the particles move based on a combination of the best point the individual particle has seen and the best point the entire swarm has seen. Efficiency is gained when one can compute a large portion of the particles simultaneously by using parallel computations. In my problem, as discussed in Section 4.2, I require parallel computation to make the cost function itself computationally feasible. As such, I desire an algorithm that seeks to minimize evaluations of the cost function and does not need to evaluate the cost function in parallel to be feasible.

Bayesian optimization [68] is a common technique for global optimization when the cost function is expensive to evaluate. The idea is to construct a probabilistic surrogate model – that is, an interpolating function that also quantifies the uncertainty one has in the estimated values between function evaluations – and pair that with a so-called acquisition function. Having a probabilistic representation of the surrogate model is key because one would like to explore areas of the optimization domain that not only have a desirable value of the cost function, but also a high un-

certainty in the actual value of the cost function; in doing so, one might "strike gold" and find a minimizer amidst an area of large interpolation uncertainty.

The workflow of Bayesian optimization is as follows:

1. Begin with surrogate model trained on N points $\{\theta_i\}_{i=1}^N$.
2. Find the minimizer θ_{N+1} of the acquisition function (which is based on the surrogate model).
3. Evaluate the cost function at θ_{N+1}.
4. Construct the set $\{\theta_i\}_{i=1}^N \cup \{\theta_{N+1}\}$ and retrain the surrogate model with the new point.

The key elements are defining the surrogate model and the acquisition function. I state my selections for these two components first and then go into detail about each afterwards.

First, regarding the choice of surrogate model, the necessary criteria are (1) to have an interpolating function such that I know what the cost function is at evaluated points, (2) to have a general trend connecting evaluated points, (3) some notion of noise between the interpolation points, and (4) an easy way to compute important test statistics such as the mean value and variance of an interpolated value. The most common way to approach this is through a Gaussian process (GP) [23, 66, 93]. The idea with a GP is to evaluate the interpolated function at a number of points. Then a *kernel* is defined that defines a trend between points as well as some distribution between the points, for example a covariance measure that allows zero uncertainty at evaluated points and maximizes uncertainty at the midpoint between evaluated points. More details may be found in Section 4.3.1, in which I use a GP as the surrogate model with the Matérn $5/2$ kernel [73]; see (4.33).

Second, the choice of acquisition function is a modeling decision based on what one finds important. There are a few common choices, but the idea is to have a function of the trend and variance of the surrogate model. For this application, I choose to employ the *expected improvement plus* acquisition function. Refer to Section 4.3.2 for a more detailed discussion of this choice and other options.

The acquisition function then needs to be optimized during the Bayesian optimization algorithm. Generally, this simply means using some form of global optimization. The specific algorithm is not important in this context, because Bayesian optimization typically assumes the evaluation of the cost function is substantially more expensive than evaluating the acquisition function. I use the MATLAB implementation of Bayesian optimization in my work herein and their implementation evaluates the acquisition function at a few thousand points and then performs local minimization on several of the "better" points. See the MATLAB documentation for the function BAYESOPT for more details [74].

Now that I have described my choices for the necessary ingredients of Bayesian optimization, I will describe GPs and acquisition functions in more detail for clarity to my decision-making process and as a reference to how each choice operates.

4.3.1 Surrogate model

The general set up of a surrogate model is that one has a set of points and then the evaluations of those points. More formally, and in the common language of the uncertainty quantification community, one has a given *experimental design* $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$ and corresponding model responses $\boldsymbol{\mathcal{Y}} = \{y_1, \ldots, y_N\} = \{\mathcal{M}(\boldsymbol{\theta}_1), \ldots, \mathcal{M}(\boldsymbol{\theta}_N)\}$, where $\mathcal{M}$ is some computational model – e.g., some function involving the solution to a set of differential equations – which one can evaluate at will given a point $\boldsymbol{\theta}$ in its domain, albeit typically at significant computational cost. In this chapter, when I apply the techniques I describe herein, my computational model $\mathcal{M}$ is my cost function $\mathcal{C}(\theta)$ described in (4.11).

The question, then, is: *how can I interpolate my model evaluations?* In a simplified setting where $\theta \in \mathbb{R}$, I might simply fit a polynomial of degree $N-1$ through $\boldsymbol{\Theta}$ and $\boldsymbol{\mathcal{Y}}$ for a global interpolation function. Or, perhaps, I want to mitigate the large oscillations that often come with high degree interpolation polynomials, so I employ piecewise polynomial interpolation or cubic spline interpolation to maintain differentiability. These methods work very well in their *deterministic* setting when the only important result of interpolation is the best guess at what the model response is.

However, in the context of Bayesian optimization, I require information regarding how confident I am about the interpolated value to encourage exploration of the optimization parameter space. For this reason, probabilistic interpolation techniques are necessary.

The idea of Gaussian process modeling, then, is to develop a probabilistic model of the form

$$\widehat{Y}(\boldsymbol{\theta})_{\text{pr}} = \boldsymbol{f}(\boldsymbol{\theta})^{\intercal}\boldsymbol{\beta} + \sigma^2 Z(\boldsymbol{\theta}) \tag{4.13}$$

as described in [93]. The first term in (4.13) is an analogue to a polynomial fit one might employ in the deterministic setting, where $\boldsymbol{f}(\boldsymbol{\theta})$ are a collection of K arbitrary functions and $\boldsymbol{\beta}$ is the vector of weights assigned to each of those functions. The inner product $\boldsymbol{f}(\boldsymbol{\theta})^{\intercal}\boldsymbol{\beta}$ is called the *trend function* in this context. Next, σ^2 denotes the constant variance assigned to the Gaussian process. Finally, $Z(\boldsymbol{\theta})$ is a zero-mean, unit-variance Gaussian process defined by an correlation function R that I will describe later. Then, following the methodology in [93], a prediction Y_0 based on parameters $\boldsymbol{\theta}_0$ and the observations $\mathcal{Y}$ follow the joint distribution

$$\begin{pmatrix} Y_0 \\ \boldsymbol{\mathcal{Y}} \end{pmatrix} \sim N_{N+1}\left[\begin{pmatrix} \boldsymbol{f}_0^{\intercal}\boldsymbol{\beta} \\ \boldsymbol{F}\boldsymbol{\beta} \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & \boldsymbol{r}_0^{\intercal} \\ \boldsymbol{r}_0 & \boldsymbol{R} \end{pmatrix}\right], \tag{4.14}$$

where $\boldsymbol{f}_0^{\intercal}\boldsymbol{\beta}$ is the trend as in (4.13), $\boldsymbol{F}\boldsymbol{\beta}$ is the trend evaluated at each point in the experimental design, $\boldsymbol{r}_0$ is the vector

$$\boldsymbol{r}_0 = \begin{pmatrix} R(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1) & \cdots & R(\boldsymbol{\theta}_0, \boldsymbol{\theta}_N) \end{pmatrix}^{\intercal}, \tag{4.15}$$

and $\boldsymbol{R}$ is the matrix

$$\boldsymbol{R} = \begin{pmatrix} R(\boldsymbol{\theta}_1, \boldsymbol{\theta}_1) & \cdots & R(\boldsymbol{\theta}_1, \boldsymbol{\theta}_N) \\ \vdots & \ddots & \vdots \\ R(\boldsymbol{\theta}_N, \boldsymbol{\theta}_1) & \cdots & R(\boldsymbol{\theta}_N, \boldsymbol{\theta}_N) \end{pmatrix}. \tag{4.16}$$

Now, I want a *predictor* of Y_0 that is optimal in some sense. The standard approach is to view this through a Bayesian perspective where (4.13) is taken as the prior model and I want to condition on the fact that $\boldsymbol{\mathcal{Y}}$ are observed. To address "optimal in some sense", one chooses the predictor $\widehat{Y}_0$ to be the *best linear unbiased predictor* meaning:

- linear: $\widehat{Y}_0 = \boldsymbol{a}_0^\intercal \boldsymbol{\mathcal{Y}}$,
- unbiased: $\mathbb{E}\left[\widehat{Y}_0 - Y_0\right] = 0$,
- best: $\widehat{Y}_0 = \min_{Y_0^* \text{ linear, unbiased}} \mathbb{E}\left[(Y_0^* - Y_0)^2\right]$.

In other words: $\widehat{Y}_0$ is a linear combination of $\boldsymbol{\mathcal{Y}}$, which is nice because the linear combination of Gaussian random variables is a Gaussian random variable; $\widehat{Y}_0$ has expectation equal to that of the true model evaluation; $\widehat{Y}_0$ has the minimum variance of all other linear, unbiased estimators.

Under this framework, one can then show that

$$\mu_{\widehat{Y}_0} = \mathbb{E}\left[\widehat{Y}_0 | \boldsymbol{\mathcal{Y}}\right] = \boldsymbol{f}_0^\intercal \widehat{\boldsymbol{\beta}} + \boldsymbol{r}_0^\intercal \boldsymbol{R}^{-1}\left(\boldsymbol{\mathcal{Y}} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right), \tag{4.17}$$

where

$$\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{F}^\intercal \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}^\intercal \boldsymbol{R}^{-1} \boldsymbol{\mathcal{Y}}. \tag{4.18}$$

Moreover, one can find that the variance of this predictor is

$$\sigma^2_{\widehat{Y}_0} = \mathbb{E}\left[\left(\widehat{Y}_0 - Y_0\right)^2\right] = \sigma^2\left(1 - \boldsymbol{r}_0^\intercal \boldsymbol{R}^{-1} \boldsymbol{r}_0 + \boldsymbol{u}_0^\intercal \left(\boldsymbol{F}^\intercal \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{u}_0\right), \tag{4.19}$$

where

$$\boldsymbol{u}_0 = \boldsymbol{F}^\intercal \boldsymbol{R}^{-1} \boldsymbol{r}_0 - \boldsymbol{f}_0. \tag{4.20}$$

See [93, Chapter 3] and [23, Chapter 1] for further details and proofs of the above. The predictor $\widehat{Y}_0$ has a few nice properties that I will outline below.

Interpolation

$\widehat{Y}_0$ *interpolates* the observed model responses through its mean value. To see this, recall (4.17) and (4.18). Then I consider one of the observed responses Y_i from parameters $\boldsymbol{\theta}_i$. Let $\boldsymbol{f}_i \equiv \boldsymbol{f}(\boldsymbol{\theta}_i)$ and $\boldsymbol{r}_i \equiv \boldsymbol{r}(\boldsymbol{\theta}_i)$. Then note:

$$\boldsymbol{r}_i^\intercal \boldsymbol{R}^{-1} = \boldsymbol{e}_i^\intercal, \tag{4.21}$$

where $\boldsymbol{e}_i^\intercal$ is the unit vector with a one in the ith row and zeros elsewhere;

$$\boldsymbol{e}_i^\intercal \boldsymbol{\mathcal{Y}} = Y_i \equiv \mathcal{M}(\boldsymbol{\theta}_i), \tag{4.22}$$

where Y_i is the ith observed model response; and

$$\boldsymbol{e}_i^\intercal \boldsymbol{F} = f_i^\intercal. \tag{4.23}$$

From these facts, I can show

$$\begin{aligned}
\mu_{\widehat{Y}_i} &= \boldsymbol{f}_i^\intercal \widehat{\boldsymbol{\beta}} + \boldsymbol{r}_0^\intercal \boldsymbol{R}^{-1} \left(\boldsymbol{\mathcal{Y}} - \boldsymbol{F}\widehat{\boldsymbol{\beta}} \right) \\
&= \boldsymbol{f}_i^\intercal \widehat{\boldsymbol{\beta}} + \boldsymbol{e}_i^\intercal \left(\boldsymbol{\mathcal{Y}} - \boldsymbol{F}\widehat{\boldsymbol{\beta}} \right) \\
&= \boldsymbol{f}_i^\intercal \widehat{\boldsymbol{\beta}} + \boldsymbol{e}_i^\intercal \boldsymbol{\mathcal{Y}} - \boldsymbol{e}_i \boldsymbol{F}\widehat{\boldsymbol{\beta}} \\
&= \boldsymbol{f}_i^\intercal \widehat{\boldsymbol{\beta}} + Y_i - \boldsymbol{f}_i^\intercal \widehat{\boldsymbol{\beta}} \\
&= Y_i.
\end{aligned}$$

This is the minimal property desired of a surrogate model: if a surrogate model cannot reproduce the (assumed to be exactly known!) model responses, then it is not a very useful model. Under a probabilistic framework, however, another fundamental notion that should be satisfied is that there is zero uncertainty surrounding the observed values. Next, I show that the predictor $\widehat{Y}_0$ satisfies this property.

Collapsing variance

While I want to regard a prediction between observed values as a random variable with some non-zero spread, the observed values themselves should reduce to a deterministic value. That is, since I actually *observed* that the model evaluated to a specific value – and I *assume* zero or negligible error since I am evaluating a deterministic computational model or have evidence that my Monte Carlo integration is converged – then I *know* what the value is. Since $\widehat{Y}_0$ is a Gaussian random variable, it suffices to show that the variance $\sigma^2_{\widehat{Y}_0} = 0$ when evaluating $\theta_i \in \boldsymbol{\Theta}$.

First, consider $\boldsymbol{u}_0$ as defined in (4.20). I will denote $\boldsymbol{u}_i$ here to emphasize it is evaluated for one of the points in the experimental design $\boldsymbol{\Theta}$. Also recall from (4.21) and (4.23) that $\boldsymbol{R}^{-1}\boldsymbol{r}_i = \boldsymbol{e}_i$ and $\boldsymbol{F}^\intercal \boldsymbol{e}_i = \boldsymbol{f}_i$. Thus, I have

$$\begin{aligned}
\boldsymbol{u}_i &= \boldsymbol{F}^\intercal \boldsymbol{R}^{-1}\boldsymbol{r}_i - \boldsymbol{f}_i \\
&= \boldsymbol{F}^\intercal \boldsymbol{e}_i - \boldsymbol{f}_i \\
&= \boldsymbol{f}_i - \boldsymbol{f}_i \\
&= 0.
\end{aligned}$$

Now, the variance is reduced to

$$\begin{aligned}
\sigma^2_{\widehat{Y}_i} &= \sigma^2 \left(1 - \boldsymbol{r}_i^\intercal \boldsymbol{R}^{-1}\boldsymbol{r}_i\right) \\
&= \sigma^2 \left(1 - \boldsymbol{r}_i^\intercal \boldsymbol{e}_i\right).
\end{aligned}$$

As a preview to the discussion on the correlation function, I see that $\boldsymbol{r}_i^\intercal \boldsymbol{e}_i = R(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i)$. As such, if $R(\boldsymbol{\theta}, \boldsymbol{\theta}) = 1$ for arbitrary, relevant $\boldsymbol{\theta}$, then I have the result I desire. Thus, this will be the condition imposed for a function R to be considered appropriate in this setting. This condition is reasonable because the correlation between a point and itself is one.

Hence, at a point in the experimental design $\boldsymbol{\theta}_i$, the predictor follows

$$\widehat{Y}_i \sim N(Y_i, 0)$$

and is thus deterministic.

Moving my attention to points outside of the experimental design, I can utilize the fact that the predictor follows a Gaussian distribution to construct confidence intervals.

Confidence intervals

One nice statistic describing the uncertainty of a value is a confidence interval. This measure provides a spread of reasonable values that one might see, based on a specified confidence level α, so one might make a more informed decision based on the context of the problem. In my case, since $\widehat{Y}_0$ is Gaussian, it is very simple to compute confidence intervals. The confidence interval of value Y_0 for parameters $\boldsymbol{\theta}_0$ is

$$Y_0 \in \left[\mu_{\widehat{Y}_0} - \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)\sigma_{\widehat{Y}_0}, \mu_{\widehat{Y}_0} + \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)\sigma_{\widehat{Y}_0}\right], \tag{4.24}$$

where Φ^{-1} is the inverse cumulative distribution function of the standard normal distribution. With a probability $1 - \alpha$, true value of Y_0 is expected to be within the given confidence interval.

I have explained properties of a GP when it is being used as an interpolating function. However, as I discussed in Section 4.2, computing the cost function to a high accuracy is too computationally expensive in my case with Monte Carlo integration. Next, I present modifications to (4.17) and (4.19) to account for noisy model evaluations.

Noisy model evaluations

I have noise in the evaluation of my cost function and I assume the noise is homoscedastic; that is, the variance of the noise is constant with respect to θ. Then, the joint distribution of the

observations and a prediction is modified from (4.14) to

$$\begin{pmatrix} Y_0 \\ \boldsymbol{\mathcal{Y}} \end{pmatrix} \sim N_{N+1}\left[\begin{pmatrix} \boldsymbol{f}_0^\intercal \boldsymbol{\beta} \\ \boldsymbol{F}\boldsymbol{\beta} \end{pmatrix}, \begin{pmatrix} \sigma^2 & \sigma^2 \boldsymbol{r}_0^\intercal \\ \sigma^2 \boldsymbol{r}_0 & \sigma^2 \boldsymbol{R} + \sigma_n \boldsymbol{I} \end{pmatrix}\right], \tag{4.25}$$

where σ_n is the noise in the cost function evaluations and $\boldsymbol{I}$ is the identity matrix. Then, the mean and variance of a prediction can be computed with [88]

$$\mu_{\widehat{Y}_0} = \mathbb{E}\left[\widehat{Y}_0 | \boldsymbol{\mathcal{Y}}\right] = \boldsymbol{f}_0^\intercal \widehat{\boldsymbol{\beta}} + \widetilde{\boldsymbol{r}}_0^\intercal \widetilde{\boldsymbol{R}}^{-1}\left(\boldsymbol{\mathcal{Y}} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right), \tag{4.26}$$

$$\sigma^2_{\widehat{Y}_0} = \mathbb{E}\left[\left(\widehat{Y}_0 - Y_0\right)^2\right] = \sigma^2_{\text{total}}\left(1 - \widetilde{\boldsymbol{r}}_0^\intercal \widetilde{\boldsymbol{R}}^{-1} \widetilde{\boldsymbol{r}}_0 + \boldsymbol{u}_0^\intercal \left(\boldsymbol{F}^\intercal \widetilde{\boldsymbol{R}}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{u}_0\right), \tag{4.27}$$

where

$$\sigma^2_{\text{total}} = \sigma^2 + \sigma_n^2, \tag{4.28}$$

$$\widetilde{\boldsymbol{r}}_0 = \left(1 - \frac{\sigma_n^2}{\sigma^2_{\text{total}}}\right)\boldsymbol{r}_0, \tag{4.29}$$

$$\widetilde{\boldsymbol{R}} = \left(1 - \frac{\sigma_n^2}{\sigma^2_{\text{total}}}\right)\boldsymbol{R} + \frac{\sigma_n^2}{\sigma^2_{\text{total}}}\boldsymbol{I}. \tag{4.30}$$

Then I can compute confidence intervals in the same manner as (4.24).

Now that the nice properties of a Gaussian process have been discussed, I want to discuss the choice of correlation function – or kernel – that finishes the construction of the Gaussian process.

Correlation function

The correlation function is a way for the designer of the Gaussian process to inject knowledge of how parameters $\boldsymbol{\theta}_0$ relate to the experimental design $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}$. For example, one might suspect that being further away – according to some norm – from the observations would result in behavior less similar to the observations and, as such, more uncertainty on the predicted response to $\boldsymbol{\theta}_0$. There is also a concept of stationarity versus non-stationarity with correlation functions. The difference, essentially, is a stationary correlation function only depends on the distance between two

points. Conversely, a non-stationary correlation function allows for differences based on where in the domain two points are. In the absence of better knowledge of the 3-step mechanisms behavior with respect to parameters θ, I take the Occam's razor approach and use the simpler choice of assuming stationarity and thus will focus on discussing the popular stationary correlation functions. The following will discuss one-dimensional correlation functions and the extension to the multivariate case will be described afterwards.

Among the most popular stationary correlation functions are the *exponential*

$$R_e(\theta_1, \theta_2; \ell) = \exp\left(-\frac{|\theta_1 - \theta_2|}{\ell}\right), \tag{4.31}$$

Gaussian (or squared exponential)

$$R_g(\theta_1, \theta_2; \ell) = \exp\left(-\frac{1}{2}\left(\frac{|\theta_1 - \theta_2|}{\ell}\right)^2\right), \tag{4.32}$$

and *Matérn* functions [73]

$$R_m(\theta_1, \theta_2; \ell, \nu) = \frac{1}{2^{\nu-1}\Gamma(\nu)}\left(2\sqrt{\nu}\frac{|\theta_1 - \theta_2|}{\ell}\right)^{\nu} \mathcal{K}_\nu\left[2\sqrt{\nu}\frac{|\theta_1 - \theta_2|}{\ell}\right], \tag{4.33}$$

where $\nu \geq 1/2$, Γ is the gamma function that generalizes factorials $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$, and $\mathcal{K}_\nu$ is the modified Bessel function of the second kind. (4.33) is rather complicated but can fortunately be greatly simplified when $\nu = p + 1/2$ for $p \in \mathbb{N}$, see [88, eq. 4.16]. The most important correlation functions in the *Matérn* family are

$$\begin{aligned} R_m(\theta_1, \theta_2; \ell, \nu = 3/2) &= \left(1 + \sqrt{3}\frac{|\theta_1 - \theta_2|}{\ell}\right)\exp\left[-\sqrt{3}\frac{|\theta_1 - \theta_2|}{\ell}\right] \\ R_m(\theta_1, \theta_2; \ell, \nu = 5/2) &= \left(1 + \sqrt{5}\frac{|\theta_1 - \theta_2|}{\ell} + \frac{5}{3}\left(\frac{|\theta_1 - \theta_2|}{\ell}\right)^2\right)\exp\left[-\sqrt{5}\frac{|\theta_1 - \theta_2|}{\ell}\right], \end{aligned} \tag{4.34}$$

as can be found in [88, eq. 4.17]. Note that $R_m(\theta_1, \theta_2; \ell, \nu = 1/2) = R_e(\theta_1, \theta_2; \ell)$, in the limit $\nu \to \infty$ $R_m(\theta_1, \theta_2; \ell, \nu) \to R_g(\theta_1, \theta_2; \ell)$, and $\nu \geq 7/2$ (including $\nu \to \infty$), are pragmatically

the same for practical applications due to the amount of smoothness (i.e., derivatives) ν imposes, see [88, ch. 4.2].

Note two problems that now arise:

1. Which of (4.31), (4.32), and (4.34) should I choose?

2. How do I choose the unknown parameter ℓ that arises in (4.31), (4.32), and (4.34)?

I answer these questions next.

Choice of correlation function

Among other possible correlation functions that I did not discuss, [88,93] describe exponential, Gaussian, and Matérn correlation functions as the most popular used in the literature. As such, there is precedence for using any of them. In [105, ch. 6.5], there is an argument for the Matérn class to be the canonical correlation function. This argument relates to allowing the parameter search to be over both ℓ and ν, as well as the choice of ν controlling smoothness; i.e., (4.31) is not continuously differentiable, (4.32) is infinitely differentiable, but (4.33) has $\lceil \nu \rceil - 1$ derivatives [88, 93]. However, [88] notes that the Matérn functions for $\nu \geq 7/2$ all behave similar to the Gaussian correlation function, which [105] argues is inappropriate.

Then, the choices are narrowed down to the Matérn functions in (4.34) and the exponential function in (4.31) (i.e., the Matérn function with $\nu = 1/2$). In [93, ch. 3.3], an empirical study is performed on how to choose the correlation function (and how to choose the parameter(s) in the correlation function; see the next discussion). Essentially, numerous methods were tested on a problem where an analytic solution is available and the error was calculated as a measure of performance. As can be seen in [93, Figures 3.6–3.8], the exponential and Matérn functions perform similarly. The authors of [93] conclude with a recommendation of using the exponential correlation function[2], but allude to evidence that the Matérn family is just as good but simply

[2]Technically, the exponential *family* is recommended, which includes $\exp(|\theta_1 - \theta_2)^p/\ell)$ for $0 < p \leq 2$ – hence also including the Gaussian correlation function – but for $0 < p < 2$ this entire family is continuous, but not continuously differentiable everywhere, so they exhibit the same smoothness properties as the $p = 1$ case I mentioned in (4.31) and thus I lump them together.

more computationally expensive. However, this study did not use the convenient representation of Matérn functions such as those in (4.34), instead using various integer values for ν for which the expensive formula (4.33) is required. The half-integer Matérn correlation functions have similar computational expense to the exponential correlation functions, so I do not consider the computation time of a Matérn function to be relevant. Because of the results in the empirical study, the recommendation to limit the smoothness of the correlation function, and the recommendation to treat the Matérn family as the canonical correlation function, I decide to use Matérn with $\nu = 5/2$ in my application. I choose $\nu = 5/2$ instead of $\nu = 3/2$ because it is the most common kernel used in the literature.

The value of the shape parameter ℓ needs to be selected in a judicious manner. This is typically done through maximum likelihood or cross-validation methods, see [93, chapter 3] for a discussion on these techniques.

It is now clear that with a Gaussian process I can compute the trend of an interpolated point conditioned on the observations through (4.26), I can extract information about the uncertainty through (4.27) and (4.24), and I can do these in an efficient manner. As discussed when I introduced Bayesian optimization, I want to sequentially build my surrogate model of the cost function by choosing new observations in a "clever manner." This "clever manner" means combining the trend of the Gaussian process with its uncertainty to search for new points to evaluate. This combination is called the acquisition function, which I discuss next.

4.3.2 Acquisition function

In a typical optimization problem – say one solved by gradient descent – the cost function is unambiguously the correct measure of how well the optimization parameters perform. However, in the context of Bayesian optimization, since a GP is the function being optimized during each iteration, the addition of uncertainty provides room for Bayesian decision making in the selection of the next evaluation point. This notion is called the *acquisition function* in the context of

Bayesian optimization. The idea is simple: choose a function of the GP that combines the trend and uncertainty in some way and optimize that.

In principle, the simplest acquisition function one might use is a *risk-free* function. That is,

$$\mathcal{A}_{\text{RF}}(\theta) = \mathbb{E}(\widehat{Y}(\theta)); \tag{4.35}$$

in other words, the acquisition function is the trend of the GP. For brevity, let me denote the mean and standard deviation of the GP evaluated at a point θ as

$$\mu(\theta) = \mathbb{E}(\widehat{Y}(\theta)), \sigma(\theta) = \sqrt{\mathbb{E}\left(\left[\widehat{Y}(\theta) - \mathbb{E}(\widehat{Y}(\theta))\right]^2\right)}. \tag{4.36}$$

That being said – particularly when few evaluations are informing the GP – exploring areas of the parameter space where uncertainty is high can yield more optimal values of the cost function. A first natural extension of (4.35) is the *lower confidence bound* acquisition function

$$\mathcal{A}_{\text{LCB}}(\theta) = \mu(\theta) - \beta\sigma(\theta); \tag{4.37}$$

that is, the GP trend minus β times the standard deviation. One might take another approach and directly calculate the probability of the cost function yielding a better value. This thought leads to the *probability of improvement* acquisition function. Since a GP provides a Gaussian distribution for observations, probability of improvement is calculated as

$$\mathcal{A}_{\text{POI}}(\theta) = \Phi\left(\frac{\arg\min_{\theta}\left[\mu(\theta)\right] - \mu(\theta)}{\sigma(\theta)}\right), \tag{4.38}$$

where Φ is the CDF of the normal distribution with mean zero and unit variance. Finally, the final acquisition function I will discuss takes the probability of improvement a step further through weighting by how much the cost function would improve. This is the *expected improvement* [58]

acquisition function given by

$$\begin{aligned}\mathcal{A}_{\text{EI}}(\theta) &= \int_{-\infty}^{\infty} \max\left(0, \arg\min_{\theta}\left[\mu(\theta)\right] - \widehat{Y}\right) \mathcal{N}\left[\widehat{Y}; \mu(\theta), \sigma^2(\theta)\right] d\widehat{Y} \\ &= \left(\arg\min_{\theta}\left[\mu(\theta)\right] - \mu(\theta)\right) \Phi\left(\frac{\arg\min_{\theta}\left[\mu(\theta)\right] - \mu(\theta)}{\sigma(\theta)}\right) \\ &\quad + \sigma(\theta)\mathcal{N}\left(\frac{\arg\min_{\theta}\left[\mu(\theta)\right] - \mu(\theta)}{\sigma(\theta)}\right),\end{aligned} \tag{4.39}$$

where $\mathcal{N}(Y)$ is the PDF of the standard normal distribution and $\mathcal{N}(Y; \mu, \sigma^2)$ is the PDF of the normal distribution with mean μ and variance σ^2. Expected improvement is one of the most popular acquisition functions and the philosophy of choosing one's next evaluation point on where one *expects* the best result resonates with me. Hence, I decide on expected improvement as my acquisition function.

From practical experience, one can find that through the course of Bayesian optimization that a certain portion of parameter space is focused on before the rest of parameter space is adequately explored. This is called over-exploiting an area. The expected improvement algorithm can be modified to detect when over-exploitation is occurring and increase the correlation function in an effort to bias towards areas of parameter space that are under-explored; see the MATLAB documentation [75] for an explanation of their implementation based on the work in [15]. By MATLAB's notation, this is called the expected improvement plus acquisition function and is what I use in my work.

Now that I have explained how Bayesian optimization works, I will apply this technique to my study of nanoparticle formation and the goal of trying to find experimental conditions that permit narrow PSDs centered around a requested size.

4.4 Optimization of experimental conditions

Now I am ready to apply Bayesian optimization to my nanoparticle problem. Recall, the goal is to optimize the multi-objective cost function (4.11). First, I will describe the optimization pa-

rameters θ I have at my disposal. Then, I will walk through some examples of optimizing over $\theta \in \mathbb{R}$ to develop intuition about what each component of the cost function affects in the optimized result. From this intuition, I will then describe my choices for the weights w_i in (4.11) and proceed to perform optimization over multiple experimental conditions simultaneously.

4.4.1 Optimization parameters

The optimization parameters that I have at my disposal can be grouped into two categories. The first group are optimization parameters where the modification of the experiment has a direct analogue within my mathematical model. These parameters are:

- A_0 – the initial precursor concentration,
- t_{end} – the time at which the reaction is stopped,
- POM_0 – the initial POM concentration.

The second group are parameters that represent changes that could be made by modifications to the experiment, but more data needs to be collected to understand exactly what experimental conditions correspond to what parameter values. These "hypothetical" parameters are:

- α_{k_f} – multiplier to the k_f reaction rate – this could be achieved by diluting the solvent ($\alpha_{k_f} < 1$), using a different solvent that acts as a stronger catalyst ($\alpha_{k_f} > 1$), or changing the ligand from POM to some other molecule;
- α_T – multiplier to the k_1, k_2, and k_3 reaction rates – this could be done by adjusting the temperature the reaction is conducted at, but the temperature dependence of these reaction rates is presently unknown.

Now that I have described the optimization parameters, in the next section I will provide visual examples of what the Bayesian optimization process looks like and what effect each component of the cost function (4.11) has.

4.4.2 Optimization visualization

I start with a visualization of the Bayesian optimization process. Recall, Bayesian optimization is composed of: (1) forming a probabilistic surrogate model of the cost function; and (2) applying an acquisition function to the surrogate model to choose the next evaluation point. In order to plot the GP surrogate model and the acquisition function as clearly as possible, I perform a one-dimensional optimization on the optimization parameter $t_{\text{end}} \in [0.1, 100]\text{h}$. Figure 4.6 shows the GP and acquisition function after five iterations and fifteen iterations. The GP plots in the left column show the trend along with two standard deviations of uncertainty. With few evaluations of the cost function, there is a lot of uncertainty about the trend. As more evaluations occur, the trend predicts the cost function much more accurately and the variance in the predictor decreases. The acquisition function plotted in the right column of Figure 4.6 also demonstrates this behavior. With few function evaluations – hence large variance in the predictor – the acquisition function takes large values nearby the current best evaluated point and other areas of parameter space. With more function evaluations and a more accurate GP, the acquisition function takes on large values only near the minimizer.

Next, I want to explore appropriate values w_i for the cost function (4.11). I sample six random values from $A_0 \in [1 \times 10^{-4}, 1]\text{molL}^{-1}$ and $t_{\text{end}} \in [0.1, 100]\text{h}$, evaluate what each component of the cost function is for those points, and plot the PSDs to see if the values I find are representative of a "good" or "bad" value. In Table 4.2 I consolidate the values of each component of the cost function and in Figure 4.7 I plot the corresponding PSDs.

For EMD, the values are on the order of 1–2 and I would say all of the PSDs are far from the "desired size" that I calculate EMD with respect to. That being said, the interpretation for EMD is quite simple: the simulated mean particle size is $EMD\text{nm}$ away from the desired particle size. I want to normalize the cost function so that each component is ≈ 1 for a "bad" PSD. $1\,\text{nm}$ apart in mean value is not a good result, so I take the EMD weight to be $w_1 = 1$. Note that I also test the effect of $w_1 = 5$ in Section 4.4.3 and explain why therein. Now considering EPS, the values in Table 4.2 are ≈ 0.5 and I consider all of the PSDs in Figure 4.7 to be wide. I take the EPS weight

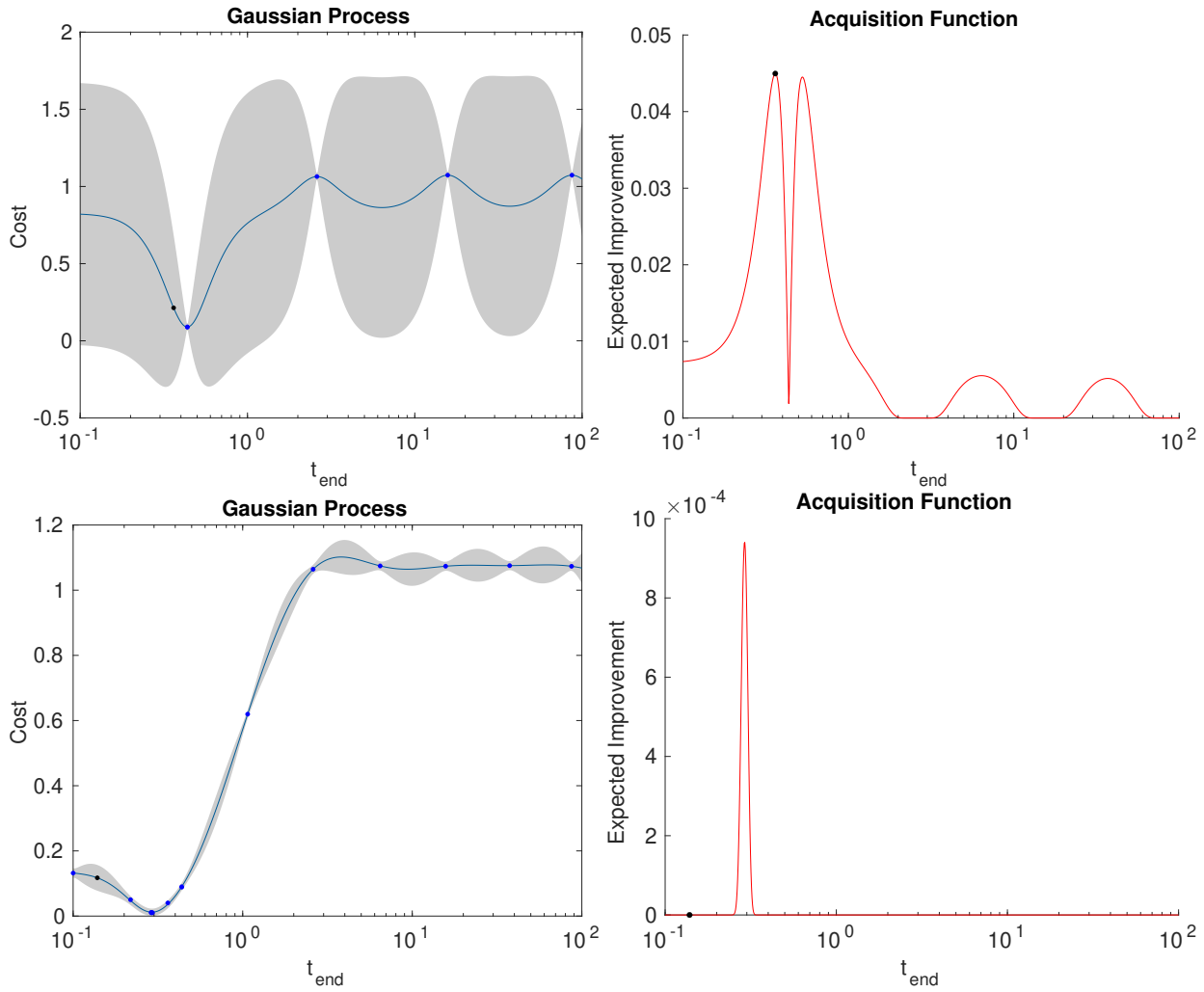


Figure 4.6: *Examples of the Bayesian optimization process after (top) 5 iterations and (bottom) 15 iterations. The left column shows the trained Gaussian process where the blue line is the trend, the blue dots are computed evaluations of the cost, and the gray shows two standard deviations of the distribution around the trend. The right column shows the evaluation of the acquisition function on the trained Gaussian process. In both plots, the black dot indicates where the next evaluation in the Bayesian optimization algorithm takes place. In the bottom plot of the acquisition function, the black dot is not located on the peak of the expected improvement curve due to the "plus" algorithm used to avoid over-sampling a small region; see [15,75] for details. Also note the difference in scale of the two plots on the right.*

Table 4.2: *Values of each component of the cost function* (4.11) *evaluated at six random points within* $A_0 \in [1 \times 10^{-4}, 1]\text{molL}^{-1}$ *and* $t_{end} \in [0.1, 100]\text{h}$. *These values compare to the PSDs in Figure 4.7.*

Function	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6
EMD	1.8	1.1	1.0	1.1	1.5	1.3
EPS	4.5×10^{-1}	7.7×10^{-1}	4.4×10^{-1}	7.7×10^{-1}	5.1×10^{-1}	5.0×10^{-1}
VMD	1.1×10^{-3}	4.7×10^{-5}	4.2×10^{-4}	4.8×10^{-5}	6.5×10^{-4}	6.7×10^{-4}
VPS	1.4×10^{-4}	1.4×10^{-5}	8.6×10^{-5}	1.4×10^{-5}	2.2×10^{-5}	3.0×10^{-5}
EAC	7.5×10^{-13}	2.5×10^{-1}	2.2×10^{-1}	2.1×10^{-1}	4.0×10^{-12}	3.2×10^{-9}

to be $w_2 = 1$ to put more emphasis on the mean value of the PSD, although a larger value such as $w_2 = 2$ would also be reasonable. Next, for VMD, values are on the order of 1×10^{-5} to 1×10^{-3}. VMD is related to the uncertainty, so I analyze the uncertainty in the PSDs in Figure 4.7. I find that the first and second PSDs are good examples of high and low uncertainty, respectively. These two PSDs correspond to VMD on the order of 1×10^{-3} for high uncertainty and 1×10^{-5} for low uncertainty. To normalize the high uncertainty case to $w_3 VMD \approx 1$, I take $w_3 = 1 \times 10^3$. Then with VPS, my analysis is similar. For the same high and low uncertainty PSDs I find values on the order of 1×10^{-4} and 1×10^{-5}, respectively. Thus I choose $w_4 = 1 \times 10^4$ so that $w_4 VPS \approx 1$ in the high uncertainty case. Finally, for EAC, the interpretation of this function is very intuitive: the reaction is $(1 - EAC) \times 100\%$ complete. It is much more efficient materials-wise to perform a reaction to its natural completion in order to produce more particles. Therefore I test both $w_5 = 2$ and $w_5 = 5$ to normalize a 50% complete and 80% complete reaction to the same cost as a PSD whose mean is $1\,\text{nm}$ from the desired mean.

Now that I have selected the weights w_i in (4.11), I present a few optimizations on the parameters $A_0 \in [1 \times 10^{-4}, 1]\text{molL}^{-1}$ and $t_{\text{end}} \in [0.1, 100]\text{h}$ to show the effect of each component of the cost function. Note that I only perform 25 iterations for each of the following cases to save on computational cost since I simply want to show the effect of each cost measure here. Analyzing Figure 4.8, I make the following observations:

1. EMD only – the only important factor is that the mean diameter in the PSD is close to the desired size, so it is no surprise that A_0 and t_{end} are found that closely match the desired size, albeit at the expense of the reaction not running to completion;

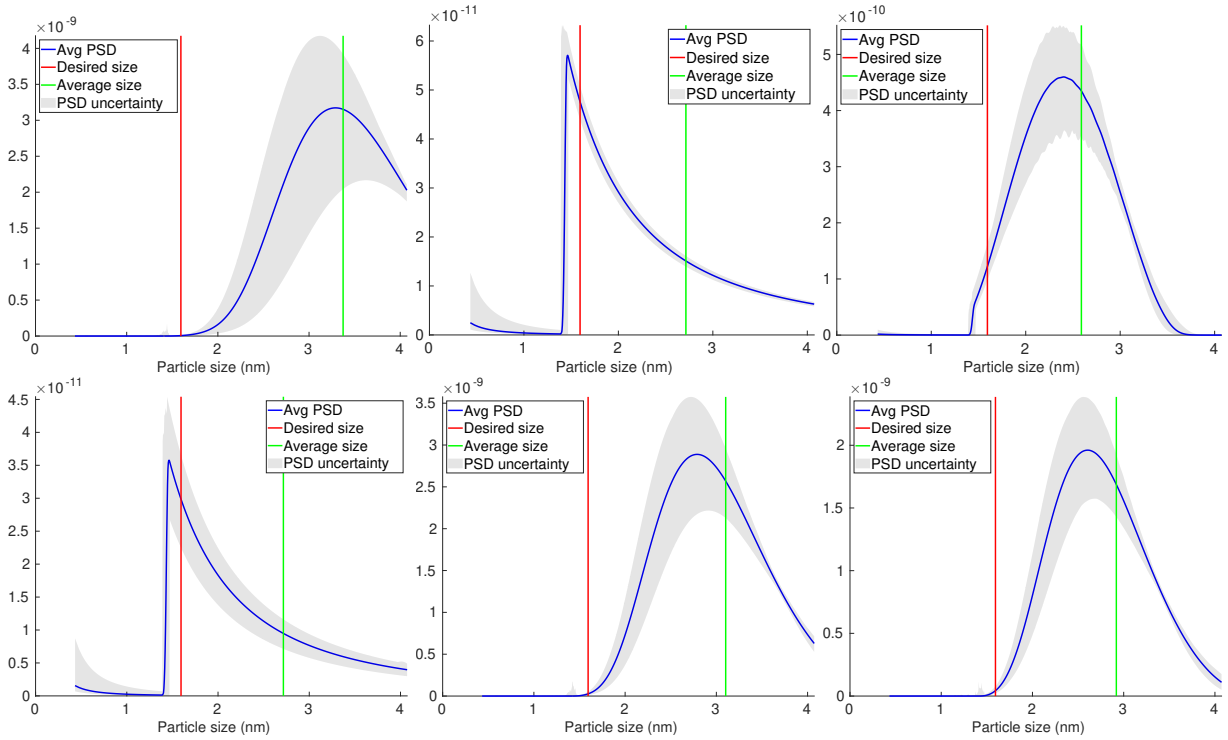


Figure 4.7: *PSDs for six random points within* $A_0 \in [1 \times 10^{-4}, 1]\,\mathrm{mol\,L^{-1}}$ *and* $t_{end} \in [0.1, 100]\,\mathrm{h}$ *in order to assess typical values of each component of the cost function* (4.11). *These figures correspond to Table 4.2 with the top row corresponding to points* 1–3 *and the bottom row to points* 4–6.

2. EMD and EPS – cutting a reaction short corresponds to lower PSD width because the reaction has not had time to create a variety of different particle sizes, so including EPS may need to be balanced by penalizing an incomplete reaction by including EAC;

3. EMD and VMD – uncertainty is penalized in this case but, as Figure 4.8 shows, uncertainty concentrated about the peak of the PSD may not influence the mean size too much and therefore be tolerated by VMD;

4. EMD and VPS – uncertainty again is penalized, except this combination results in a qualitatively different PSD than the previous examples, indicating that the variance of the PSD spread may be more sensitive to uncertainty than the PSD mean;

5. EMD and EAC – this combination penalizes an incomplete reaction and, as Figure 4.8 demonstrates, the cost function interprets it as desirable to have a complete reaction whose PSD mean is far from the requested size, and hence EAC needs to be balanced by the other cost components;

6. All cost component – the five functions that form (4.11) form a balance where the PSD mean is close to the desired size, the uncertainty is relatively low, and the reaction is mostly completed.

Now I am ready to optimize the PSD shape and perform this analysis in the next section.

4.4.3 Optimization results

In this section I find that I am able to affect the shape of the PSD to target a specific size particle. I am not able to create a *monodisperse* PSD through my optimization[3], but I will demonstrate the ability to find experimental conditions where my simulations predict a narrow PSD centered around a size close to my desired size. For the results I present herein, I fix my desired size at 150 atoms which corresponds to $1.59\,\mathrm{nm}$. In the following, I test the effects of two different ways to weight

[3]Nor is a *truly* monodisperse PSD expected in nature from a multi-step, self-assembly process!

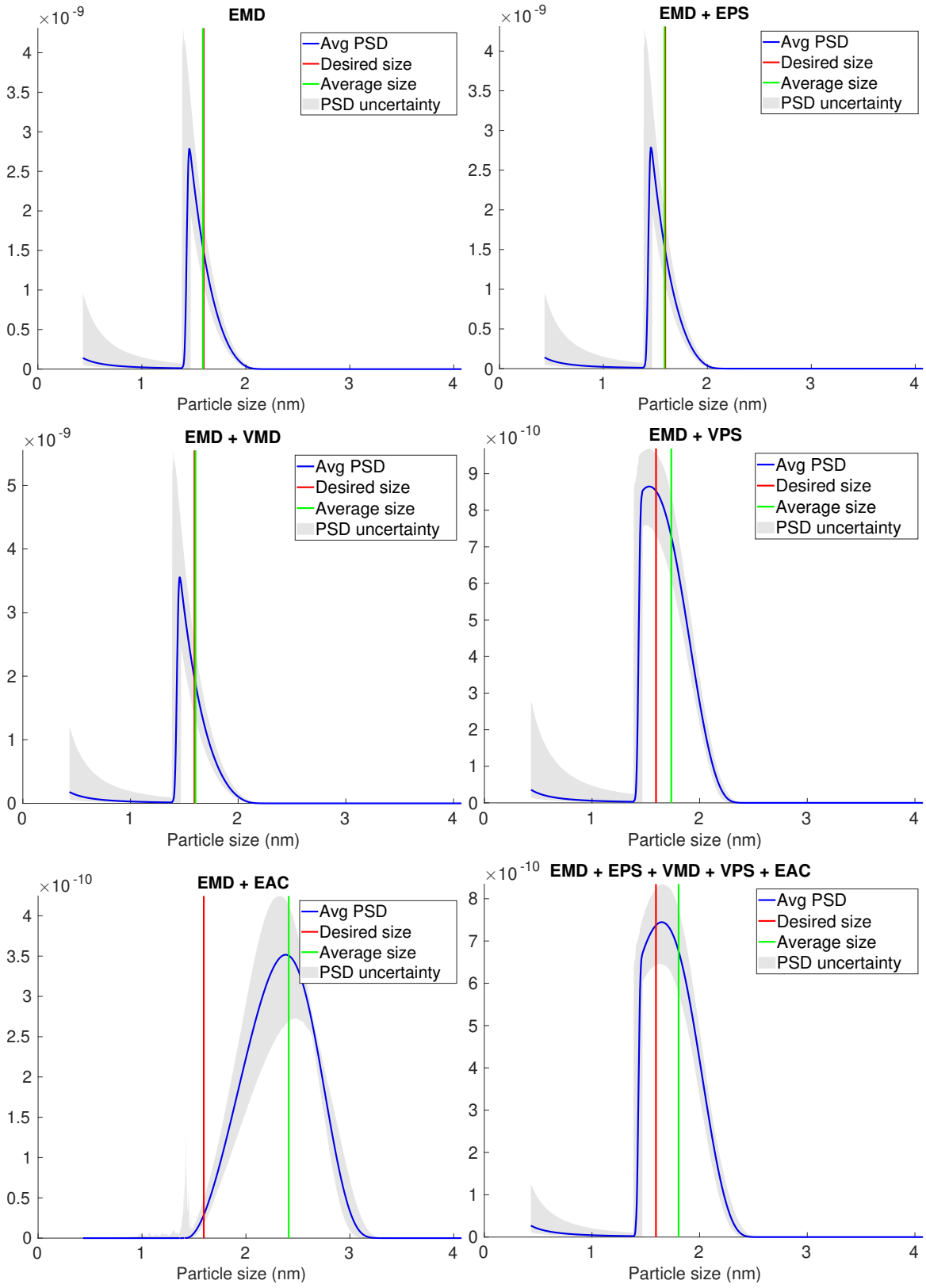


Figure 4.8: *PSDs resulting from the (incomplete) optimization of parameters* $A_0 \in [1 \times 10^{-4}, 1]$ *and* $t_{end} \in [0.1, 100]$ *for subsets of the components of the cost function* (4.11).

the cost function (4.11). The first way follows the analysis I detailed in the previous section:

$$\begin{aligned} \boldsymbol{w}^{(1)} &= \begin{bmatrix} w_1, & w_2, & w_3, & w_4, & w_5 \end{bmatrix} \\ &= \begin{bmatrix} 1, & 1, & 1 \times 10^3, & 1 \times 10^4, & 2 \end{bmatrix}. \end{aligned} \tag{4.40}$$

I will denote this weighting as "risk averse" in the figures I present in this section. The second weighting methodology puts more emphasis on matching the mean value and completing a complete reaction and disregards the uncertainty to simulate a "high risk, high reward" mentality:

$$\begin{aligned} \boldsymbol{w}^{(2)} &= \begin{bmatrix} w_1, & w_2, & w_3, & w_4, & w_5 \end{bmatrix} \\ &= \begin{bmatrix} 5, & 1, & 0, & 0, & 5 \end{bmatrix}. \end{aligned} \tag{4.41}$$

I will denote this weighting as "risk tolerant" in the figures I present in this section. I then perform five optimizations:

1. Optimization over parameters A_0, t_{end}, and POM_0 with the weights $\boldsymbol{w}^{(1)}$;
2. Optimization over parameters A_0, t_{end}, and POM_0 with weights $\boldsymbol{w}^{(2)}$ and weakening the stopping condition on the Bayesian optimization algorithm to allow at minimum 250 iterations;
3. Optimization over parameters A_0, t_{end}, POM_0, α_{k_f}, and α_T with weights $\boldsymbol{w}^{(1)}$;
4. Optimization over parameters A_0, t_{end}, POM_0, α_{k_f}, and α_T with weights $\boldsymbol{w}^{(2)}$;
5. Optimization over parameters A_0, t_{end}, POM_0, α_{k_f}, and α_T with weights $\boldsymbol{w}^{(2)}$ and removing the stopping condition on the Bayesian optimization algorithm.

For the first four optimizations I perform, I enforce a stopping condition on the Bayesian optimization algorithm. After 50 iterations (250 for the second; I explain this later in this section), I start tracking the ratio of the largest expected improvement to the best observed cost function value. Once the expected improvement drops below 0.1% of the optimal cost value, then I end

my optimization routine. I do, however, cap the total number of evaluations of the cost function at 500. This is a simple heuristic I decided on for the purposes of this analysis; see [79] for a detailed discussion around this stopping criterion. Bayesian optimization is often prescribed to simply stop after a set amount of time or number of iterations. Recently, there has been increased attention towards developing more robust stopping criterion, for example the work in [71], but comparing different stopping criterion is out of the scope of my work. For the fifth optimization, I noticed that the other optimization analyses stopped well short of the maximum number of iterations I allow. Thus, I perform a fifth optimization analysis where I simply perform 500 iterations of Bayesian optimization to see if my results are noticeably better than when I use my simple stopping criterion. Now I will discuss the results of each of these optimizations grouped by the optimization parameters included.

First, I optimize with respect to A_0, t_{end}, and POM_0. The optimized PSDs are shown in Figure 4.9. The first optimization is performed with weights $\boldsymbol{w}^{(1)}$. When I conduct this optimization, I find the optimal parameters are $A_0 = 4.7892 \times 10^{-2}\,\text{molL}^{-1}$, $t_{\text{end}} = 9.9728 \times 10^{1}\,\text{h}$, $POM_0 = 1.059 \times 10^{-4}\,\text{molL}^{-1}$, and the average A utilization is 92.8%. The reaction is mostly complete based on how much precursor is used in the optimized experiment, but the optimized PSD is far from what I am looking for[4] as can be seen by how much the PSD mean deviates from the desired particle size in the left plot in Figure 4.9. When I switch to using $\boldsymbol{w}^{(2)}$, the results reverse. From the optimized parameters $A_0 = 1.018 \times 10^{-4}\,\text{molL}^{-1}$, $t_{\text{end}} = 2.815\,\text{h}$, and $POM_0 = 1.0015 \times 10^{-4}\,\text{molL}^{-1}$, I find that the PSD mean lines up almost exactly with the desired mean as is shown in the right plot in Figure 4.9. However, the reaction is cut well short based on the average A utilization being 16.9%, hence the reaction is relatively wasteful. In both of these cases, I suspect a higher computational budget might lead me to a better PSD. This, however, is a limitation of both Bayesian optimization and optimization in general when the cost function is expensive to evaluate and therefore must be considered when performing an analysis like the one I

[4]This reasoning is based on the *average* particle size being far from the desired size, but note that the *mode* of the PSD is close to the desired size. A potential future research project would be to investigate these optimizations by measuring the central tendency of a PSD by the highest concentration particle rather than the average particle size.

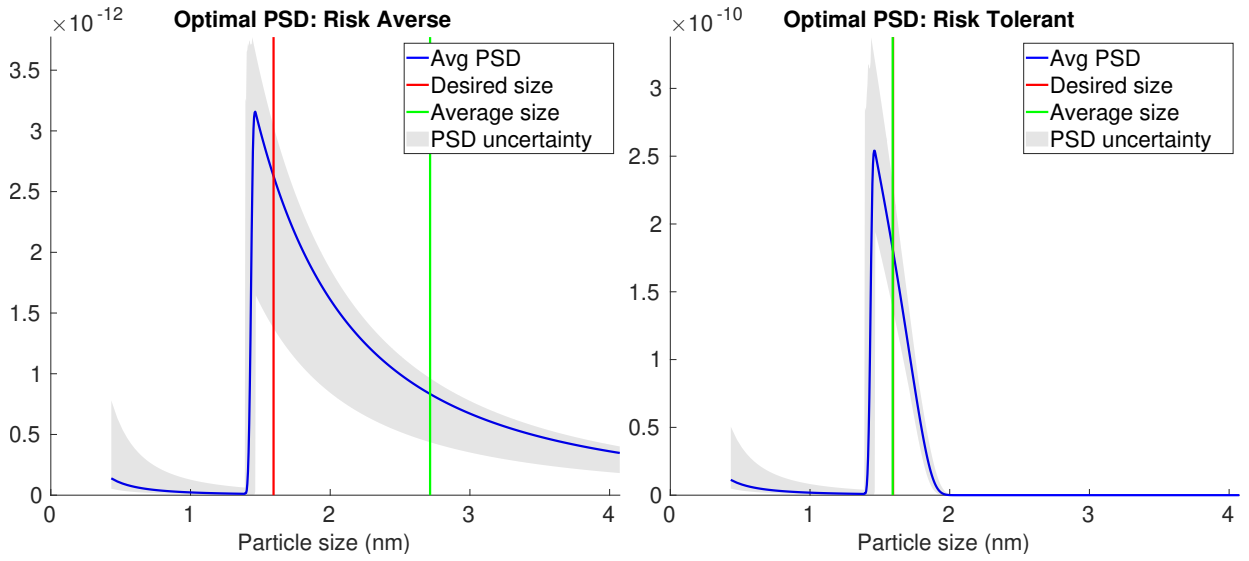


Figure 4.9: *Optimal PSDs when optimizing the experimental conditions t_{end}, A_0, and POM_0. (Left) The optimal PSD when weights* (4.40) *are used. (Right) The optimal PSD when weights* (4.41) *are used.*

conduct here. Future research could explore computing derivatives of the cost function (4.11) and supplying the Bayesian optimization algorithm with gradient information [24, 119].

Now, I add α_{k_f} and α_T as optimization parameters. I plot the resulting PSDs in Figure 4.10. First, using weights $\boldsymbol{w}^{(1)}$, I find the optimal parameters are: $A_0 = 5.5825 \times 10^{-4}\,\mathrm{molL}^{-1}$, $t_{\text{end}} = 5.0759\,\mathrm{h}$, $POM_0 = 1.0368 \times 10^{-4}\,\mathrm{molL}^{-1}$, $\alpha_{k_f} = 9.9295$, and $\alpha_{k_T} = 5.0266 \times 10^{-1}$. The resulting PSD is the left plot in Figure 4.10 and on average 99.5% of the precursor is used up, indicating the reaction is performed to completion. The generated particles are, however, approximately $0.5\,\mathrm{nm}$ larger than desired. Similarly, when I use weights $\boldsymbol{w}^{(2)}$ the optimized PSD is strikingly similar as I show in the right plot of Figure 4.10. The optimal parameters are different: $A_0 = 1.3 \times 10^{-3}\,\mathrm{molL}^{-1}$, $t_{\text{end}} = 6.8169 \times 10^{1}\,\mathrm{h}$, $POM_0 = 1.1538 \times 10^{-4}\,\mathrm{molL}^{-1}$, $\alpha_{k_f} = 9.9612$, and $\alpha_{k_T} = 6.9087 \times 10^{-1}$. Again, the suggested reaction is performed until its natural end since the average A utilization is 100%. I did notice that I met my termination condition after 52 iterations – only 2 iterations after I allow the optimization to conclude – and I want to test if that was a premature conclusion to my Bayesian optimization. In fact, it *must* be the case that the global optimum was not found because I am optimizing over a superset of parameters compared to the previous experiment and the "risk tolerant" results in Figure 4.9 are better than in Figure 4.10. Thus, I

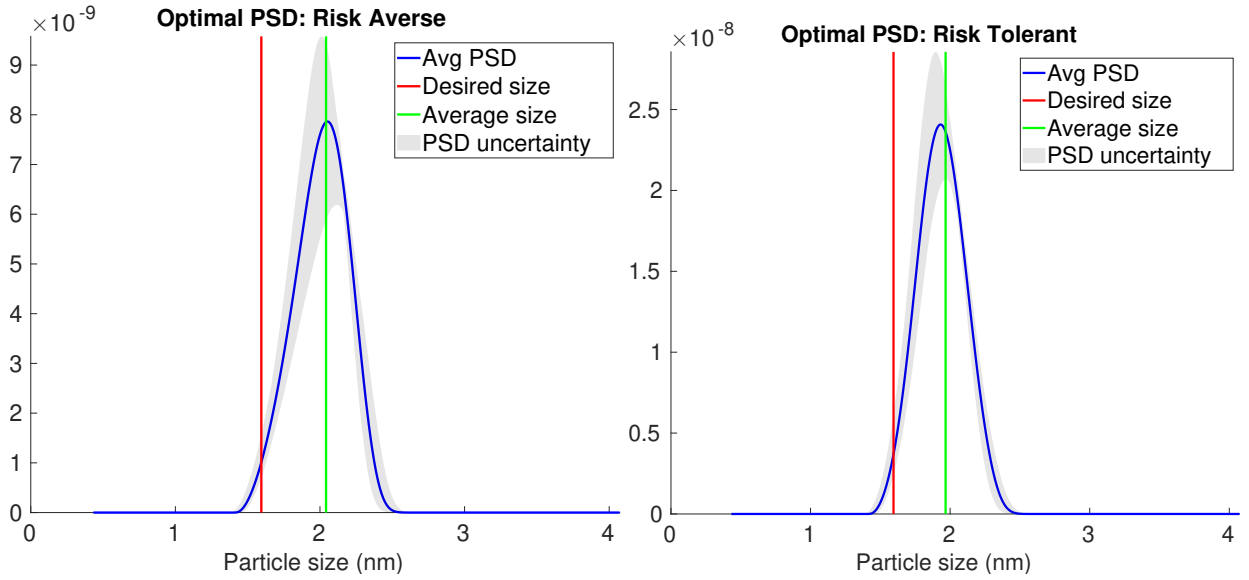


Figure 4.10: *Optimal PSDs when optimizing the experimental conditions t_{end}, A_0, POM_0, α_{k_f}, and α_T. (Left) The optimal PSD when weights* (4.40) *are used. (Right) The optimal PSD when weights* (4.41) *are used.*

eliminated the expected improvement termination condition and simply ran 500 iterations. The resulting PSD can be seen in the left plot in Figure 4.11. With optimized parameters $A_0 = 2.7667 \times 10^{-4}\,\mathrm{molL}^{-1}$, $t_{\mathrm{end}} = 8.4036 \times 10^{-1}\,\mathrm{h}$, $POM_0 = 1.0004 \times 10^{-4}\,\mathrm{molL}^{-1}$, $\alpha_{k_f} = 9.8811$, $\alpha_{k_T} = 3.9556$, I simulate an extremely narrow PSD whose mean size is very close to the size I wanted. The average A utilization is 76.5%, which means there is still some time left in the reaction, but I am still pleased with the balance between resource efficiency of this proposed experimental setup and the resultant PSD. That being said, the PSD is slightly smaller than the size I requested. The right plot in Figure 4.11 shows the resulting PSD after $2\,\mathrm{h}$ where the reaction now consumes 87.2% of the precursor. I found this by manually increasing the reaction time until I was pleased with the result. This manual fine-tuning is clearly not ideal – future research should involve a local optimization after the (global) Bayesian optimization[5] – but the point I want to express is how a local search following Bayesian optimization can improve results.

[5]For example, one could perform Bayesian optimization *only* on t_{end} after the other parameters are identified, or one could apply a local optimization algorithm like Nelder-Mead in a small neighborhood around the parameters the Bayesian optimization algorithm found.

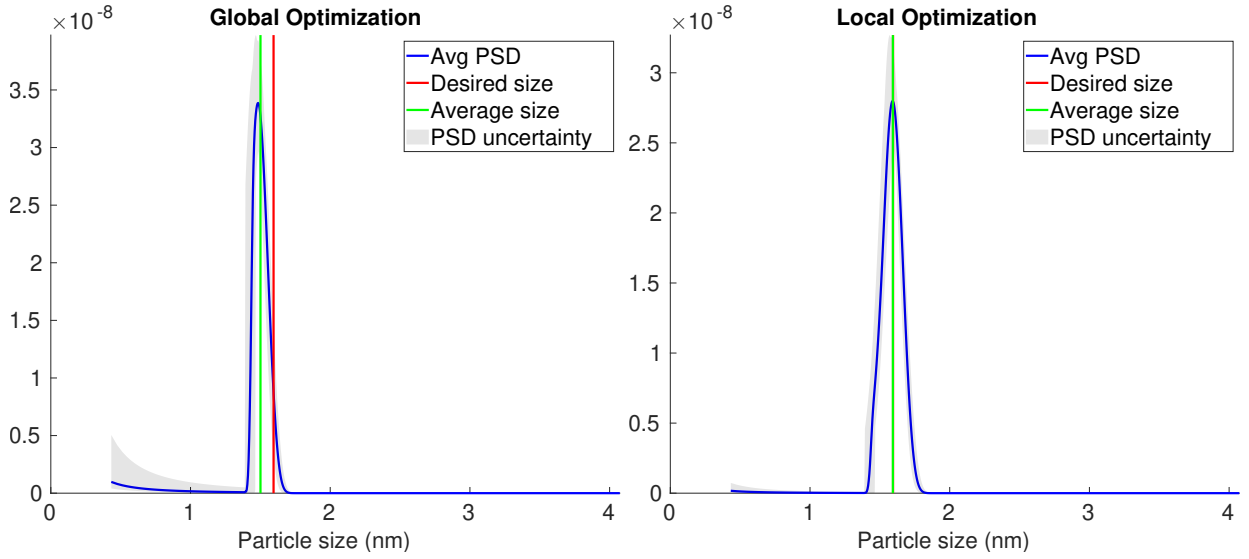


Figure 4.11: *Optimal PSDs when optimizing the experimental conditions t_{end}, A_0, POM_0, α_{k_f}, and α_T. These plots are comparable to the methodology of the right plot in Figure 4.10 except the termination condition on Bayesian optimization was weakened to stop after 500 iterations. (Left) The optimal PSD determined by Bayesian optimization. (Right) The optimal PSD determined by a local search of only t_{end} performed following Bayesian optimization.*

This chapter dealt with the treatment of optimizing the shape of the PSD in such a way that accounted for the uncertainty in the model parameters. In fact, I showed that I can systematically deduce experimental conditions that correspond to a pre-specified, desired outcome. What would be preferable, however, is to not have to worry about the uncertainty because I know the model parameters $\mathcal{K}$ to such a degree of accuracy that I can work in the deterministic setting. This, of course, occurs in the limit of having infinite data, but having a sufficient amount of data will make this approximation reasonable. The next chapter explores the question of determining what data to collect next in order to better determine the model parameters $\mathcal{K}$ and thus decrease the uncertainty in my simulations.

Chapter 5

Optimal experimental design

This chapter explores a Bayesian approach to experimental design. My goal is to design a new experiment that provides as much information as possible. I interpret this as finding experimental conditions where my simulations have high uncertainty. That is to say, I *want* to conduct an experiment under conditions where my simulations currently have high uncertainty. High uncertainty indicates I have limited information – that is, my data are not informative in those experimental conditions – and thus a judicious experiment would measure under those conditions of excessive uncertainty because this measurement could shine substantial light onto which of the parameter values is the "true" one. Depending on what I find, this could have two effects:

1. Allow me to perform Bayesian inference like I did in Chapter 3 incorporating the new data and thus learning more about the posterior distribution;

2. Explore limitations of the model I discussed in Chapter 2 by finding experimental conditions that do not align with my assumptions.

Mathematical models are, necessarily, imperfect. Hence either of these two effects are interesting to investigate: either I develop a posterior distribution that allows for more accurate simulations or I discover model limitations that provide me with information about when my model is useful or spurs further research that leads to the discovery of a more accurate model.

This chapter is organized as follows. First, I will discuss how one defines an optimal design. Then, I will outline the numerical algorithm I use to compute this optimal design. Finally, I will apply this approach to the nanoparticle problem I investigate and discuss my results.

5.1 Defining optimal experimental design

The first question is: what do I mean by an *optimal* design? If one models their data with a polynomial regression model, then there is substantial literature discussing approaches to optimal design; for example, see [17, 22] for reviews of methods.

Consider the example linear regression model

$$Y = a_0 + a_1 X, \tag{5.1}$$

where Y predicts some measurement, X is the independent variable I assume the measurement is related to, and a_0 and a_1 are model parameters that need to be inferred. Now I want to know for what values of X should I conduct my experiment to best estimate a_0 and a_1. One of the most popular methods to approach this problem is to find the so-called *D-optimal design* [102]. Suppose N candidate points $\boldsymbol{X}_N = \{X_n\}$ are requested and consider the evaluation of the linear regression model on these points

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{M}\boldsymbol{a} \\ \boldsymbol{Y} &= \begin{pmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_N \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}. \end{aligned} \tag{5.2}$$

Then, the D-optimal design $\boldsymbol{X}_N^*$ is given by

$$\boldsymbol{X}_N^* = \underset{\boldsymbol{X}_N \in \mathbb{R}^N}{\arg\min} \left(\frac{1}{\det(\boldsymbol{M}^\intercal \boldsymbol{M})} \right), \tag{5.3}$$

where $\det$ is the determinant operator. The idea with D-optimality is that the matrix $(\boldsymbol{M}^\intercal \boldsymbol{M})^{-1}$ represents the variance and covariance of the model parameters a_0 and a_1. Hence, minimizing the determinant in (5.3) is equivalent to minimizing the variability in parameter estimates (as measured

by the determinant). The minimization of (5.3) can then be conducted through an optimization algorithm. Moreover, the evaluation of $\det(\boldsymbol{M}^\intercal \boldsymbol{M})$ is very computationally efficient in the linear regression case since it is simply a 2×2 matrix. Other optimality measures such as A-optimality and G-optimality can be taken [22].

This optimal design problem on a linear regression model can be viewed through a Bayesian lens as well. From the Bayesian standpoint, let me assume the following:

- The likelihood distribution is modeled as $\pi_{\mathrm{L}}(\boldsymbol{Y}|\boldsymbol{a}, \sigma^2) = N(\boldsymbol{M}\boldsymbol{a}, \sigma^2 \boldsymbol{I})$;
- The prior distribution is $\pi_{\mathrm{pr}}(\boldsymbol{a}) = N(\boldsymbol{a}_0, \sigma^2 \boldsymbol{R}^{-1})$.

Here, σ^2 is the known variance of the observations $\boldsymbol{Y}$, $N(\cdot, \cdot)$ is the normal distribution, $\boldsymbol{R}$ is some known matrix, and $\boldsymbol{Y}$, $\boldsymbol{M}$, and $\boldsymbol{a}$ follow the same notation as (5.2). In this setting there are numerous interpretations of D-optimal, but one interpretation can be shown to be

$$\boldsymbol{X}_N^* = \underset{\boldsymbol{x}_N \in \mathbb{R}^N}{\arg\min} \left(\frac{1}{\det(\boldsymbol{M}^\intercal \boldsymbol{M} + R)} \right); \tag{5.4}$$

see [17] and references therein for a discussion of this Bayesian optimality condition along with alternatives. The reason the formulations (5.3) and (5.4) are useful choices is because the entries of the matrix $(\boldsymbol{M}^\intercal \boldsymbol{M})^{-1}$ represent the variance and covariances of the model parameters. Hence, minimizing the determinant of this matrix (or this matrix scaled by a fixed noise level) minimizes the volume of the ellipsoid in parameter space in which the parameters are likely to be. Other optimality criteria include seeking to minimize the average variance of the parameters (A-optimality) and minimizing the variance of a prediction while accounting for the number of model parameters and suggested experiments (G-efficiency) [22].

Formulations such as (5.3) and (5.4) become much more complicated when one's model is nonlinear. In particular, when the likelihood distribution (and hence the posterior) are intractable to calculate. As discussed in Chapter 3, my nanoparticle problem satisfies this intractability. To understand how to approach this more challenging case, I will introduce some *information theory*.

Mutual information [99] – also known as Kullback-Leibler (KL) divergence [67] – is a measure of the similarity of two distributions. For continuous probability distributions $p(x)$ and $q(x)$, mutual information is given by

$$D_{\text{KL}}(p\|q) = \int p(x) \ln\left(\frac{p(x)}{q(x)}\right) dx. \tag{5.5}$$

When $p(x) = q(x)$, then $D_{\text{KL}}(p\|q) = 0$. Otherwise, $D_{\text{KL}}(p\|q) > 0$. In fact, the Bayesian D-optimality condition (5.4) can be derived by using (5.5) and optimizing with respect to maximizing the information gain from the prior distribution ($q(x)$) to the posterior distribution ($p(x)$).

In the setting of optimal experimental design, I will need to pose (5.5) in a slightly different manner. Consider my knowledge base at the onset of this optimal experimental design study. I have completed a Bayesian inference problem as described in Chapter 3, so I have *developed* knowledge about the distribution of parameters $\mathcal{K}$ through the posterior distribution $\pi(\mathcal{K}|\text{data})$. Therefore, as I seek the optimal experimental design, my *current, expert knowledge* is the posterior distribution I computed in Chapter 3[6]. Recall, as I explained in Section 3.3, the prior distribution encompasses my current beliefs about the distribution of $\mathcal{K}$. Hence, in this setting, I take $\pi(\mathcal{K}) = \pi(\mathcal{K}|\text{data})$[7], where I drop the conditioning on "data" because the data analyzed in Chapter 3 is now enveloped in my current knowledge base and the *future* data I design an experiment to collect does not inform this distribution. Then I consider a proposed *design*, $\boldsymbol{d}$, – that is, experimental conditions – and data that I would collect, $\boldsymbol{y}$, if I were to perform design $\boldsymbol{d}$. Then, in theory, if I were to collect $\boldsymbol{y}$ and perform a Bayesian inference analysis, I would compute a posterior distribution $\pi(\mathcal{K}|\boldsymbol{d}, \boldsymbol{y})$. The mutual information between the posterior and prior distributions would then be computed by [63, 64, 91]

$$U(\boldsymbol{d}) = \int \pi(\mathcal{K})\pi(\boldsymbol{y}|\mathcal{K}, \boldsymbol{d}) \ln\left(\frac{\pi(\mathcal{K}|\boldsymbol{d}, \boldsymbol{y})}{\pi(\mathcal{K})}\right) d\mathcal{K}d\boldsymbol{y}, \tag{5.6}$$

[6]Specifically, the posterior distribution for the 3-step mechanism visualized in Figure 3.5 since I determined in Section 3.7 that the data-driven evidence supports the 3-step over the 4-step.

[7]Although $\pi(\mathcal{K})$ could be taken as the same prior distribution I use in Chapter 3 if data has not been collected yet.

where $\pi(\boldsymbol{y}|\mathcal{K}, \boldsymbol{d})$ is the likelihood distribution associated with $\pi(\mathcal{K}|\boldsymbol{d}, \boldsymbol{y})$ and I introduce the notation $U(\boldsymbol{d})$ to denote this is a *utility* function that I will be optimizing later on. The interpretation of (5.6) is that $U(\boldsymbol{d}) \approx 0$ indicates the design $\boldsymbol{d}$ provides little information that the prior distribution $\pi(\mathcal{K})$ did not already provide and as a result $\pi(\mathcal{K}|\boldsymbol{d}, \boldsymbol{y}) \approx \pi(\mathcal{K})$. Hence, the experiment conducted was not very useful. On the other hand, $U(\boldsymbol{d}) \gg 0$ indicates the design provides a lot of information and the resulting posterior distribution will be significantly different than the prior distribution if that data were collected.

To exactly compute (5.6), I would need to choose a design $\boldsymbol{d}$, conduct the experiment and collect data $\boldsymbol{y}$ (really, collect all possible data realizations $\boldsymbol{y}$ in order to compute the integral in (5.6)), and then perform Bayesian inference to compute the posterior distribution $\pi(\mathcal{K}|\boldsymbol{d}, \boldsymbol{y})$. Clearly, this is not realistic. Instead, the utility is approximated by [64]

$$U(\boldsymbol{d}) \approx \frac{1}{N} \sum_{i=1}^{N} \ln \left[\frac{\pi(\mathcal{K}^{(i)}|\boldsymbol{d}, \boldsymbol{y}^{(i)})}{\pi(\mathcal{K})} \right], \tag{5.7}$$

where model parameters $\mathcal{K}^{(i)}$ are drawn from the prior distribution $\pi(\mathcal{K})$ and data $\boldsymbol{y}^{(i)}$ is simulated via the computational model where design $\boldsymbol{d}$ is specified and parameters $\mathcal{K}^{(i)}$ are taken. I can draw from the prior distribution[8] and, given $\boldsymbol{d}$ and $\mathcal{K}^{(i)}$, I can simulate data by:

1. computing a PSD by solving (2.13);
2. forming a multinomial distribution from the particle concentrations;
3. drawing from that multinomial distribution M times to simulate observing M particles in a TEM image.

All that is left is computing the log-ratio

$$\ln \left[\frac{\pi(\mathcal{K}^{(i)}|\boldsymbol{d}, \boldsymbol{y}^{(i)})}{\pi(\mathcal{K})} \right]. \tag{5.8}$$

[8]Take the MCMC samples generated in Chapter 3, thin the chains as described in Section 3.6, determine the bandwidth matrix BW of a KDE applied to the thinned samples, draw a random sample s from the thinned MCMC chains, draw a random number r from $N(0, BW)$, and finally the observation is given by $s + r$.

This computation is nontrivial and [91] provides a review of modern methodologies for this computation. I decide to use a methodology introduced more recently in [109] and applied in [63, 64]. The next section discusses how to compute the log-ratio (5.8) in a tractable manner.

5.2 Computing the utility of an experimental design

The authors of [109] propose a method called Likelihood-Free Inference by Ratio Estimation (LFIRE) as a computationally efficient way of estimating (5.8). In this section, I describe the LFIRE algorithm and how to apply it to my nanoparticle problem.

At its core, the LFIRE algorithm involves simulating data observations from the prior and posterior distributions and then posing a classification problem to see if the observations from each distribution can be distinguished. The rationale is that if there is high uncertainty in the simulation for some design $\boldsymbol{d}$, then the classifier will be able to distinguish between the prior and posterior simulated data. This will then correspond to

$$\ln\left[\frac{\pi(\mathcal{K}^{(i)}|\boldsymbol{d},\boldsymbol{y}^{(i)})}{\pi(\mathcal{K})}\right] \gg 0.$$

Before discussing LFIRE, however, I need to express (5.8) in terms of likelihood (i.e., data-generating) distributions. I assume the design has no impact on my prior distribution, hence

$$\pi(\mathcal{K}|\boldsymbol{d}) = \pi(\mathcal{K}), \tag{5.9}$$

and use Bayes' theorem to note

$$\pi(\mathcal{K}|\boldsymbol{d},\boldsymbol{y}) = \frac{\pi(\boldsymbol{y}|\mathcal{K},\boldsymbol{d})\pi(\mathcal{K}|\boldsymbol{d})}{\pi(\boldsymbol{y}|\boldsymbol{d})}. \tag{5.10}$$

Then I find that (dropping the (i) in (5.8) for generality)

$$\frac{\pi(\mathcal{K}|\boldsymbol{d},\boldsymbol{y})}{\pi(\mathcal{K})} = \frac{\pi(\boldsymbol{y}|\mathcal{K},\boldsymbol{d})\pi(\mathcal{K}|\boldsymbol{d})}{\pi(\boldsymbol{y}|\boldsymbol{d})} \times \frac{1}{\pi(\mathcal{K})} \tag{5.11}$$

and apply the relationship (5.9) to conclude

$$\frac{\pi(\mathcal{K}|\boldsymbol{d},\boldsymbol{y})}{\pi(\mathcal{K})} = \frac{\pi(\boldsymbol{y}|\mathcal{K},\boldsymbol{d})}{\pi(\boldsymbol{y}|\boldsymbol{d})}. \tag{5.12}$$

Hence, I use LFIRE to approximate

$$\ln\left(\frac{\pi(\mathcal{K}|\boldsymbol{d},\boldsymbol{y})}{\pi(\mathcal{K})}\right) = \ln\left(\frac{\pi(\boldsymbol{y}|\mathcal{K},\boldsymbol{d})}{\pi(\boldsymbol{y}|\boldsymbol{d})}\right) \tag{5.13}$$

which allows me to approximate (5.6) via (5.7). The LFIRE method then proceeds as follows.

First, generate a single set of simulated data from the prior distribution. That is, sample N_{prior} model parameters from $\pi(\mathcal{K})$ and simulate *one* measurement per $\{\mathcal{K}^{(i)}\}_{i=1,\dots,N_{\text{prior}}}$ to form the set of simulated data $\mathcal{Y}_{\text{prior}} = \{\boldsymbol{y}^{(i)}\}_{i=1,\dots,N_{\text{prior}}}$. Next, sample a single parameter $\mathcal{K}^{(i)}$ from $\pi(\mathcal{K})$. This will correspond to the ith summand in (5.7), so this step is repeated N_{MC} times. With the sampled parameter $\mathcal{K}^{(i)}$, I then generate N_{post} simulated data sets to form $\mathcal{Y}_{\text{post}} = \{\boldsymbol{y}^{(i,j)}\}_{j=1,\dots,N_{\text{post}}}$.

Now that simulated data has been generated, LFIRE proposes a nonlinear logistic regression problem to attempt to classify the data $\mathcal{Y}_{\text{prior}}$ and $\mathcal{Y}_{\text{post}}$ as coming from their respective distributions. This regression problem is posed as

$$\mathbb{P}(\boldsymbol{y}_0 \in \mathcal{Y}_{\text{post}}; h) = \frac{1}{1+\nu\exp(-h(\boldsymbol{y}_0))}, \tag{5.14}$$

where $\mathbb{P}$ indicates the probability of an event, $h(\boldsymbol{y})$ is a nonlinear function, and $\nu = N_{\text{prior}}/N_{\text{post}}$. This function $h(\boldsymbol{y})$ is chosen carefully. In [109, Appendix A], the authors prove that if

$$\begin{aligned}\mathcal{J}(h,\mathcal{K},N_{\text{prior}},N_{\text{post}}) = \frac{1}{N_{\text{prior}}+N_{\text{post}}}\Bigg[&\sum_{i=1}^{N_{\text{post}}}\ln\left(1+\nu\exp\left\{-h(\boldsymbol{y}_{\text{post}}^{(i)})\right\}\right)\\ &+\sum_{i=1}^{N_{\text{prior}}}\ln\left(1+\frac{1}{\nu}\exp\left\{h(\boldsymbol{y}_{\text{prior}}^{(i)})\right\}\right)\Bigg],\end{aligned} \tag{5.15}$$

and the function $h^*(\boldsymbol{y})$ is defined as

$$h^*(\boldsymbol{y}) \coloneqq \lim_{N_{\text{post}}, N_{\text{prior}} \to \infty} \left[\arg\min_{h(\boldsymbol{y}, \mathcal{K})} \mathcal{J}(h, \mathcal{K}, N_{\text{prior}}, N_{\text{post}}) \right], \tag{5.16}$$

then $h^*(\boldsymbol{y})$, in fact, satisfies

$$h^*(\boldsymbol{y}) = \ln \left(\frac{\pi(\boldsymbol{y}|\mathcal{K}, \boldsymbol{d})}{\pi(\boldsymbol{y}|\boldsymbol{d})} \right). \tag{5.17}$$

Therefore, if I solve the limiting minimization problem (5.16) then I have also computed the log-ratio that I desire. Of course, finding the exact function $h^*(\boldsymbol{y})$ is intractable, so one uses finite sample sizes N_{prior} and N_{post} and restricts the function space of $h(\boldsymbol{y})$ to be

$$\widehat{h}(\boldsymbol{y}) = \boldsymbol{\beta}^\intercal \boldsymbol{\psi}(\boldsymbol{y}), \tag{5.18}$$

where $\boldsymbol{\psi} = [\psi_1(\boldsymbol{y}), \cdots, \psi_b(\boldsymbol{y})]$ are summary statistics such as mean and variance of the simulated data and $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_b] \in \mathbb{R}^b$. The computation of the log-ratio is now condensed to fixing N_{prior} and N_{post} and inferring $\boldsymbol{\beta}$ through the cost function

$$\begin{aligned} \widehat{\mathcal{J}}(\boldsymbol{\beta}, \mathcal{K}) = \frac{1}{N_{\text{prior}} + N_{\text{post}}} \Bigg[& \sum_{i=1}^{N_{\text{post}}} \ln \left(1 + \nu \exp \left\{ -\boldsymbol{\beta}^\intercal \boldsymbol{\psi}(\boldsymbol{y}_{\text{post}}^{(i)}) \right\} \right) \\ & + \sum_{i=1}^{N_{\text{prior}}} \ln \left(1 + \frac{1}{\nu} \exp \left\{ \boldsymbol{\beta}^\intercal \boldsymbol{\psi}(\boldsymbol{y}_{\text{prior}}^{(i)}) \right\} \right) \Bigg], \end{aligned} \tag{5.19}$$

wherein minimizing $\widehat{\mathcal{J}}(\boldsymbol{\beta}, \mathcal{K})$ with respect to $\boldsymbol{\beta}$ provides an approximation to the log-ratio (5.17). The LFIRE method proposes inferring $\boldsymbol{\beta}$ with the regularization problem

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^b} \left[\widehat{\mathcal{J}}(\boldsymbol{\beta}, \mathcal{K}) + \lambda \sum_{i=1}^{b} |\beta_i| \right]. \tag{5.20}$$

The regularization parameter λ is deduced by performing cross-validation [106]. That is, the following procedure is performed:

1. Pick $\lambda > 0$;
2. Split the simulated data into K disjoint sets $\left\{\mathcal{Y}_{\text{prior}}^{(i)}, \mathcal{Y}_{\text{post}}^{(i)}\right\}_{i=1,\dots,K}$;
3. Organize the split data into training sets $\left\{\mathcal{Y}_{\text{post}}^{\text{train}}, \mathcal{Y}_{\text{prior}}^{\text{train}}\right\}$ and testing sets $\left\{\mathcal{Y}_{\text{post}}^{\text{test}}, \mathcal{Y}_{\text{prior}}^{\text{test}}\right\}$, ensuring the training and testing sets are disjoint;
4. Using as the data $\left\{\mathcal{Y}_{\text{post}}^{\text{train}}, \mathcal{Y}_{\text{prior}}^{\text{train}}\right\}$, perform the optimization algorithm to find $\boldsymbol{\beta}_{\lambda}^{*}$ via (5.20);
5. Test the accuracy of the classification by computing probabilities of data originating from the posterior by computing (5.14) and tabulating the percentage of correct classifications: correct for $\boldsymbol{y} \in \mathcal{Y}_{\text{post}}^{\text{test}}$ means $\mathbb{P}(\boldsymbol{y} \in \mathcal{Y}_{\text{post}}^{\text{train}}) \geq 0.5$, whereas correct for $\boldsymbol{y} \in \mathcal{Y}_{\text{prior}}^{\text{test}}$ means $\mathbb{P}(\boldsymbol{y} \in \mathcal{Y}_{\text{post}}^{\text{train}}) < 0.5$;
6. Repeat steps 3–5 until all split sets $\left\{\mathcal{Y}_{\text{prior}}^{(i)}, \mathcal{Y}_{\text{post}}^{(i)}\right\}_{i=1,\dots,K}$ have been the testing set exactly once;
7. Average the correct classification percentage for each of the K cross-validation iterations to compute a *fitness score* for the current λ;
8. Repeat steps 1–7 within an optimization algorithm (e.g. Bayesian optimization) to find the optimal λ^{*} with the fitness score as the cost function.

Once λ^{*} is found, the minimization problem (5.20) is solved using all of the simulated data. With $\boldsymbol{\beta}^{*}$ found, the log-ratio (5.8) is simply computed as

$$\ln\left[\frac{\pi(\mathcal{K}^{(i)}|\boldsymbol{d}, \boldsymbol{y}^{(i)})}{\pi(\mathcal{K})}\right] \approx \boldsymbol{\beta}^{*\intercal}\boldsymbol{\psi}(\boldsymbol{y}_{0}^{(i)}), \tag{5.21}$$

where $\boldsymbol{y}_{0}^{(i)}$ is one last simulated measurement from the posterior with $\mathcal{K}^{(i)}$.

For my implementation of this algorithm, I make the following choices. I consider a simulated measurement drawing 300 particles from the multinomial distribution induced by the simulated concentrations. For the sake of computational expense, I let $N_{\text{prior}} = N_{\text{post}} = N_{\text{MC}} = 500$. I also

use 2-fold cross-validation instead of the recommended 10-fold cross-validation to save computation time. To optimize λ, I use Bayesian optimization and to optimize $\boldsymbol{\beta}$ with the MATLAB function FMINUNC, which uses the BFGS quasi-Newton minimization algorithm [14, 29, 38, 98]. Finally, for summary statistics $\boldsymbol{\psi}$, I compute the sample mean and sample variance of the simulated particle measurements for each time the design specifies. Hence, if $\boldsymbol{d}$ specifies collecting data at M different times, then

$$\boldsymbol{\psi} = \left[\mu_i, \quad \sigma_i^2, \quad \cdots \quad \mu_M, \quad \sigma_M^2\right], \tag{5.22}$$

where μ_i is the simulated PSD mean at the ith time and σ_i^2 is the simulated PSD variance at the ith time. Other summary statistics may provide better results, but that investigation is out of the scope of my work.

With the contents of this section, I now have a methodology to compute an approximation of the utility (5.7) for a single design $\boldsymbol{d}$. In order to find the optimal design, I perform Bayesian optimization with the cost function being $-U(\boldsymbol{d})$, where the negative is introduced because software implementations of optimization algorithms minimize instead of maximize by convention. For the results in the following section, I impose the same stopping conditions as in Chapter 4: after 50 iterations start tracking the expected improvement relative to the best observed objective function value, stop the optimization if expected improvement falls below 0.1%, and allow a maximum of 500 function evaluations. In the next section, I present my results of finding the optimal experimental design.

5.3 Optimal experimental design results

In this section I present three different optimal designs. The first two designs demonstrate the identification of simulations with high uncertainty. The last design demonstrates the identification of finding the limits of my mathematical model. In this section I will present these three designs and discuss my interpretations.

The first design scenario I present allows a single observation while performing the nanoparticle reactions using the same conditions as the collected data I used in Chapter 3. Recall, these conditions are: $A_0 = 1.2 \times 10^{-3}\,\mathrm{molL}^{-1}$ and $POM_0 = 0\,\mathrm{molL}^{-1}$, among other conditions such as the temperature that I omit in my discussion; see [117] for a detailed discussion of the experimental methodology. I request data to be collected at a single additional time and allow that measurement to occur within 0.1–$10\mathrm{h}$. Figure 5.1 shows the PSDs along with uncertainty for the optimal design I found and the design with the lowest mutual information that was evaluated during my optimization procedure. I find the optimal design in this case is to conduct a measurement at $0.102\,\mathrm{h}$. The optimized design exhibits exactly the elevated uncertainty that I expect. As can be seen in Figure 5.1, around the peak of the distribution, the uncertainty causes my simulations to predict concentrations from $2 \times 10^{-10}\,\mathrm{molL}^{-1}$ to $6 \times 10^{-10}\,\mathrm{molL}^{-1}$ with an average of approximately $3.5 \times 10^{-10}\,\mathrm{molL}^{-1}$. That is, roughly speaking, my simulation predicts a peak particle concentration of $3.5 \times 10^{-10} \pm 100\%$! This finding also matches my scientific intuition. There are a lot of fast dynamics within the 3-step mechanism, namely in the nucleation mechanism. Previously, the first measurement occurred after approximately one hour, so it stands to reason that a measurement early on within the reaction would help calibrate the parameters $\mathcal{K}$ to account for the early reaction kinetics that likely are smoothed out as the reaction undergoes.

On the other hand, the example of low mutual information – that is, a suboptimal design – that I present in Figure 5.1 also tells a fascinating tale. The PSD I present corresponds to the lowest mutual information I observed during optimization. This happens at $6.697\,\mathrm{h}$, which is after the reaction has completed and hence the PSD should be similar to the data collected at $4.838\,\mathrm{h}$, for which I have a relatively high number of particle observations. Notice the uncertainty is much more controlled than in the optimal design: the maximum uncertainty occurs around the peak of the distribution and I find this range is approximately $(2 \pm 0.5) \times 10^{-9}\mathrm{molL}^{-1}$, a much lower relative difference than in the optimal design. Interestingly, another design with very low mutual information (close in value to the example I plot in Figure 5.1) is taking a measurement at $0.9786\,\mathrm{h}$. Recall, the first observed data in Chapter 3 occurred at $0.918\,\mathrm{h}$ and this data set included the most

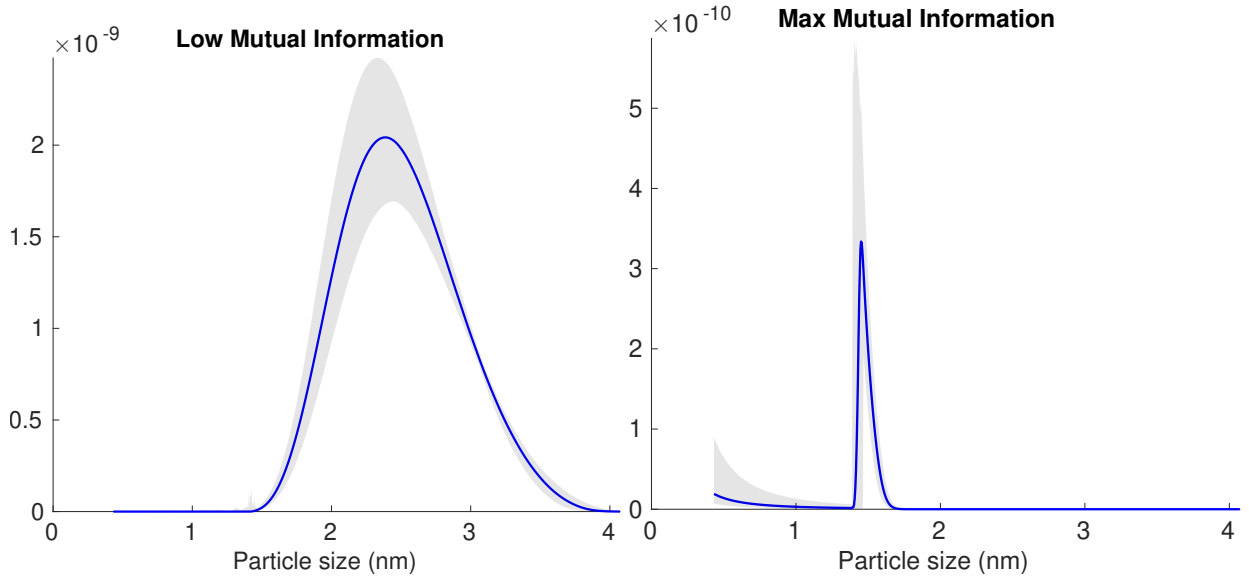


Figure 5.1: *Simulated PSD with a design (left) a measurement at a late time of* $6.697\,\mathrm{h}$*, resulting in low mutual information – i.e. a non-informative design – and (right) a measurement at an early time of* $0.102\,\mathrm{h}$*, resulting in the optimized design to maximize the mutual information.*

observed particles. My optimization methodology investigated an observation similar to where I already observed a large number of particles and concluded it is not worth observing more data there. That is precisely what should happen.

Next, I extend my design to collecting measurements at four different times throughout the reaction, while maintaining the conditions $A_0 = 1.2 \times 10^{-3}\,\mathrm{molL}^{-1}$ and $POM_0 = 0\,\mathrm{molL}^{-1}$. Figure 5.2 presents the resulting optimal design. The suggested measurement times are 0.113, 0.483, 1.158, and $4.603\,\mathrm{h}$. Similar to the optimal design presented first, a measurement at $0.113\,\mathrm{h}$ makes sense scientifically and based on the uncertainty present in the PSD. The measurement at $0.483\,\mathrm{h}$ also makes sense: another measurement early on in the reaction and the uncertainty for the smallest particle sizes is high as can be seen in Figure 5.2 for sizes around $0.5\,\mathrm{nm}$. The measurement at $1.158\,\mathrm{h}$ is intriguing. On the one hand, the PSD uncertainty appears relatively low, but on the other hand this time is very close to my data observation at $1.17\,\mathrm{h}$ which has very sparse data. It makes sense to me that a repeat observation at this point in the reaction would provide insight that is not gained because the observed PSD does not have enough particles to produce a well-defined size distribution. Finally, the observation at $4.603\,\mathrm{h}$ does not seem like the most

optimal choice. I already have a good observation close to that time and the uncertainty in the PSD appears low. I suspect I terminated my Bayesian optimization early and that including more simulated data and more samples in the Monte Carlo samples in the approximation (5.7) would help identify a more optimal design. Alternatively, my intuition may simply differ from what the mutual information criteria finds. Alternative visualizations may illuminate why $4.603\,\mathrm{h}$ is an optimal measurement, but that is a topic for further research.

Finally, I again extend my design to four separate measurements and allow for $A_0 \in [1 \times 10^{-4}, 1 \times 10^{-2}]\mathrm{molL}^{-1}$ and $POM_0 \in [1 \times 10^{-4}, 1 \times 10^{-1}]\mathrm{molL}^{-1}$. The optimized design is $A_0 = 1.5363 \times 10^{-4}\,\mathrm{molL}^{-1}$, $POM_0 = 1.8439 \times 10^{-4}\,\mathrm{molL}^{-1}$, and the measurement times are 0.039, 8.640, 12.960, and $19.659\,\mathrm{h}$. In this analysis, I allowed for the reaction to last up to $24\,\mathrm{h}$ since adjusting precursor and POM concentrations can cause the reaction to take longer. The PSDs corresponding to the measurements are in Figure 5.3. I find this optimal design extremely insightful. First off, all the PSDs in Figure 5.3 seem to have large uncertainty over a large range of particle sizes. More interesting, however, are the results of the last two suggested measurements (the bottom row of Figure 5.3). These show a very low overall concentration of particles, but the particles are accumulating at the maximum particle size I allow. Mathematically, the reason this occurs is because the high concentration of POM causes the reversible reaction in (2.2) to highly favor $\mathrm{A} \cdot \mathrm{L} + 2\,\mathrm{solv}$ as opposed to $\mathrm{A(solv)}_2 + \mathrm{L}$. Hence, the concentration of $\mathrm{A(solv)}_2$ remains low, which quashes nucleation due its quadratic dependence on $\mathrm{A(solv)}_2$. Then, the few particles that are formed exist within an abundance of the precursor A and thus are able to grow to much larger sizes according to my mathematical model. At first glance, it may seem as if the only inappropriate model assumption is that clusters do not form in sizes significantly larger than $4\,\mathrm{nm}$. However, an empirical study was performed in [115] wherein this reaction was conducted with $A_0 = 1.2 \times 10^{-3}\,\mathrm{molL}^{-1}$ and $POM_0 = 9 \times 10^{-3}\,\mathrm{molL}^{-1}$. My simulations according to my mathematical model and the experimental design results in Figure 5.3 would suggest this experiment would result in iridium nanoparticles in excess of $4\,\mathrm{nm}$ or 2500 atoms. However, the findings of [115] are that these experimental conditions result in particles containing approximately 150 iridium atoms. That

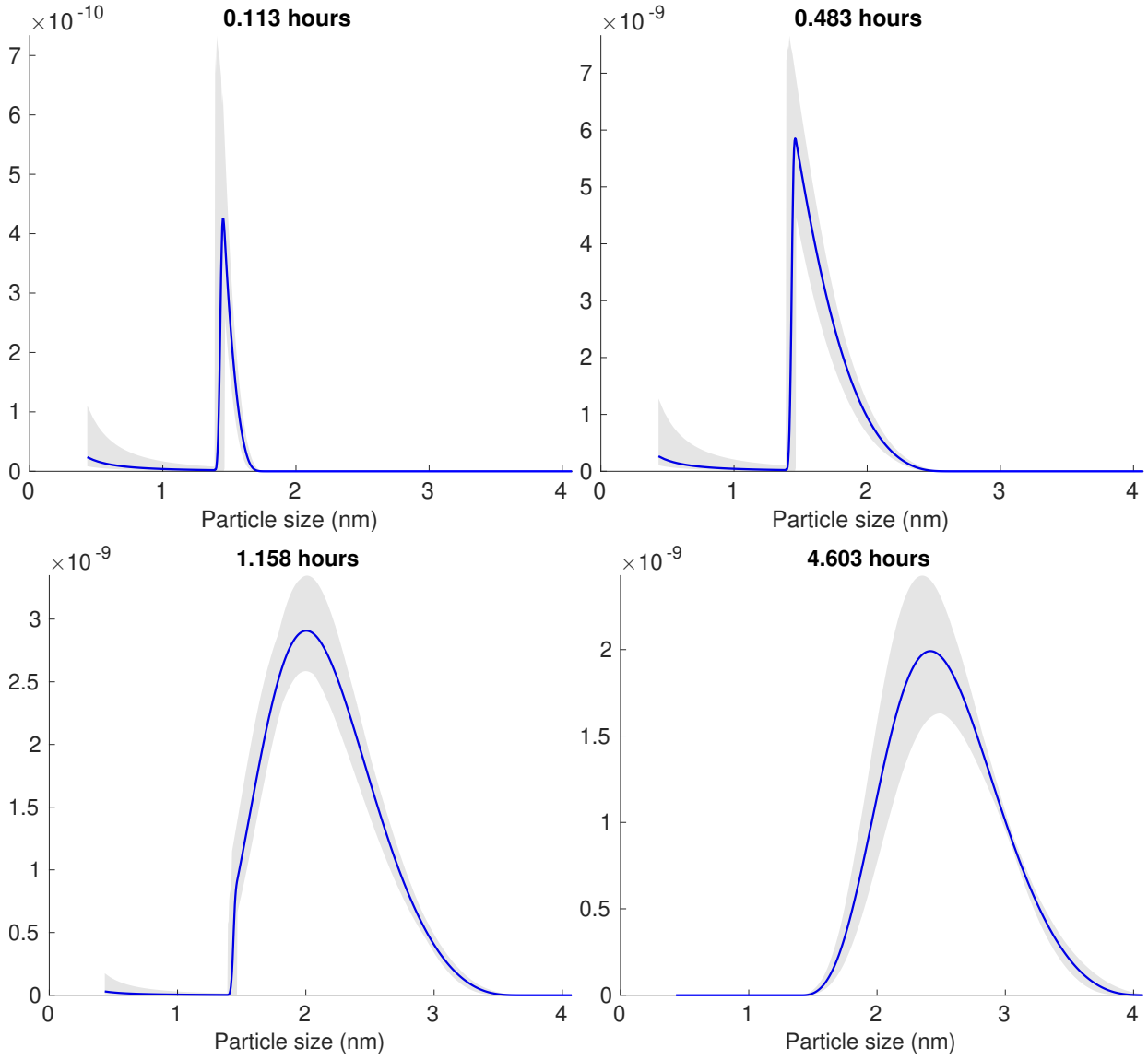


Figure 5.2: *Simulated PSDs at the optimal design times where the design allows for four measurements at four different times without changing the experimental conditions as compared to the data in Chapter 3.*

is, the increased presence of POM, as postulated by the authors of [115], causes the ligand POM to "block" the reactionary surface of iridium particles and therefore prevent them from growing larger. My simulations and the results in [115] are contradictory. This contradiction suggests an insufficiency in modeling the POM dependence of particle growth. Moreover, it suggests the need to further study the growth (and agglomeration) kernel discussed in Chapter 2 to account for the ligand-dependence of nanoparticle formation in a more robust manner. The hope, then, is that a model applicable to a larger range of experimental conditions is developed.

The results of this chapter demonstrate a methodology to deduce an experimental design that allows me to maximally extract new information about the iridium nanoparticle system from the collected data. When experimental conditions are kept static and only the measurement times are adjusted, areas of high uncertainty are identified and my results suggest I need more information about the early stage of the reaction to better understand the fast nucleation kinetics that occur. On the other hand, when experimental conditions are allowed to vary, deficiencies in the mathematical model are systematically identified, indicating that my optimal experimental design formulation is also an important tool in the *disproof*-based mechanistic description of nanoparticle formation.

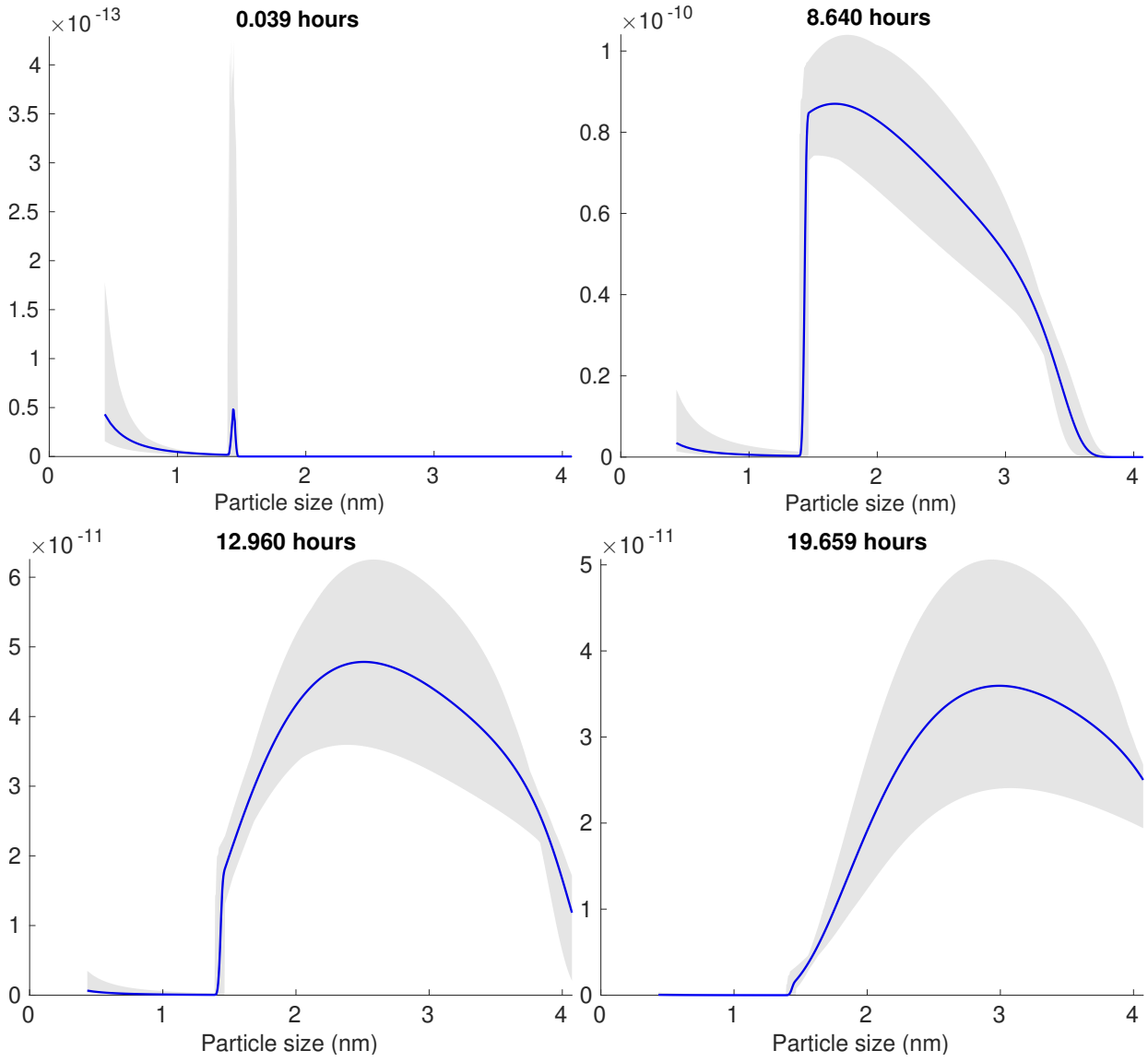


Figure 5.3: *Simulated PSDs at the optimal design where the design allows for four measurements at different times and modifications to the initial concentrations of the precursor and ligand POM.*

Chapter 6

Summary

My dissertation work fell into the intersection of nanoparticle chemistry, mathematics, and statistics. Through my approaches I was able to answer important scientific questions, which I summarize in the remainder of this chapter.

In Chapter 2, I introduced the scientific and mathematical modeling approach to mechanistic nanoparticle formation that provided the foundation of my work. This methodology was developed over a number of years by my colleagues. Most importantly, I began with two models to explore:

1. The *3-step* mechanism wherein nanoparticles form via nucleation and particle growth;

2. The *4-step* mechanism wherein nanoparticles form via nucleation, particle growth, and nucleation.

I started my contributions by analyzing the computational efficiency of solving the equations that model the nanoparticle-forming chemical reactions. I was able to utilize sparse matrices to reduce the memory requirements, a more sophisticated ODE solver to acquire higher accuracy with fewer time steps, and sparse linear algebra to solve the linear system arising at each time step more efficiently. Overall, I was able to attain a $\sim 20\times$ increase in computational efficiency through my efforts.

Next, in Chapter 3 I explored a project on estimating the parameters that arise in the models introduced in Chapter 2. Not only did I estimate optimal parameter values, but I constructed probability distributions for those parameters. In so doing, I was able to accurately characterize uncertainty in the model and had a methodology to propagate parameter uncertainty to uncertainty in the nanoparticle size distribution. I constructed the probability distributions through an iterative technique, so I detailed a thorough analysis to ensure that I converged to the true probability distribution throughout this iterative process. After developing probability distributions for two different models of nanoparticle formation, I approached the problem of quantitatively selecting

the more appropriate model by using information within the probability distributions surrounding the parameters of each model. At the conclusion of this project, I was able to:

- Find parameters that optimally correspond to the experimental data for both the 3-step and 4-step mechanisms;
- Construct *accurate* probability distributions of the model parameters in order to *accurately* capture uncertainty for both the 3-step and 4-step mechanisms;
- Quantitatively determine that there is substantially more data-driven evidence that the 3-step mechanism is a more appropriate model than the 4-step mechanism, thus answering the scientific questions of whether agglomeration occurs within the iridium nanoparticle system I study herein.

With the conclusion of this project, my results naturally brought me to a new investigation, which was the subject of my next project.

In Chapter 4, the characterization of model parameters led me to ask if I could now *control* the nanoparticle reaction to realize a desired particle size distribution. In order to answer this question, I developed a mathematical representation of what a desirable particle size distribution is, while incorporating the uncertainty inherent in the model due to parameter uncertainty. From this step, I tackled the challenges of an optimization problem without access to derivatives and an expensive function to evaluate at each iteration. Using Bayesian optimization, I was able to circumvent these issues to find experimental setups that resulted in a nanoparticle-forming chemical reaction that produced nanoparticles in a near-monodisperse distribution centered around a requested average size. As the properties of nanoparticles are dependent on their size in many applications, my work has provided a systematic framework for the *crucial* scientific challenge of controlling particle size distributions. The role of uncertainty present in this project and my previous one motivated my final project through a desire to reduce the parameter uncertainty.

My final project in Chapter 5 explored how to better characterize model parameters by efficiently conducting a new experiment to collect data. As performing these nanoparticle-forming

chemical reactions and the subsequent data collection can be labor, time, and resource intensive, having a computational approach to designing experimental conditions is, again, a scientific question of significant interest. In this project, I broach the seemingly intractable question of optimizing the expected change in the probability distribution of model parameters when data is collected according to a to-be-optimized experimental design. To accomplish my goal, I utilized a specialized algorithm developed in the statistics community to deduce an optimal experimental design in a computationally feasible time frame. I proposed three optimal designs satisfying different experimental constraints. Two of these designs matched with scientific intuition by proposing taking measurements early in the reaction when I know that my current data does not capture the fast chemical kinetics that occur early in the nanoparticle-forming chemical reactions, which provided strong evidence of the usefulness of my approach. The last design identified shortcomings in the assumptions I made when selecting the 3-step mechanism as the mathematical model, which is also an interesting and useful effect my approach has. Thus, this project provides a means to understand nanoparticle-forming chemical reactions in a resource efficient manner by designing optimal experimental conditions.

My dissertation explored three projects at the cusp of modern nanoparticle science. Characterizing accurate mathematical models, controlling particle size distributions, and knowing what experiment to conduct next are preeminent scientific questions that are actively researched and I made progress in techniques to accomplish all three questions.

Bibliography

[1] J. D. AIKEN, Y. LIN, AND R. G. FINKE, *A perspective on nanocluster catalysis: polyoxoanion and* $(\text{n-}C_4H_9)_4N^+$ *stabilized* $Ir(0)_{\sim 300}$ *nanocluster 'soluble heterogeneous catalysts'*, Journal of Molecular Catalysis A: Chemical, 114 (1996), pp. 29–51.

[2] N. ARMSTRONG AND D. HIBBERT, *An introduction to Bayesian methods for analyzing chemistry data*, Chemometrics and Intelligent Laboratory Systems, 97 (2009), pp. 194–210.

[3] M. R. AXET AND K. PHILIPPOT, *Catalysis with Colloidal Ruthenium Nanoparticles*, Chemical Reviews, 120 (2020), pp. 1085–1145.

[4] T. BAYES, *LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S*, Philosophical Transactions of the Royal Society of London, 53 (1763), pp. 370–418.

[5] I. E. BECK, V. I. BUKHTIYAROV, I. Y. PAKHARUKOV, V. I. ZAIKOVSKY, V. V. KRIVENTSOV, AND V. N. PARMON, *Platinum nanoparticles on* Al_2O_3*: Correlation between the particle size and activity in total methane oxidation*, Journal of Catalysis, 268 (2009), pp. 60–67.

[6] R. D. BERRY, H. N. NAJM, B. J. DEBUSSCHERE, Y. M. MARZOUK, AND H. ADALSTEINSSON, *Data-free inference of the joint distribution of uncertain model parameters*, Journal of Computational Physics, 231 (2012), pp. 2180–2198.

[7] C. BESSON, E. E. FINNEY, AND R. G. FINKE, *A Mechanism for Transition-Metal Nanoparticle Self-Assembly*, Journal of the American Chemical Society, 127 (2005), pp. 8179–8184.

[8] ——, *Nanocluster Nucleation, Growth, and Then Agglomeration Kinetic and Mechanistic Studies: A More General, Four-Step Mechanism Involving Double Autocatalysis*, Chemistry of Materials, 17 (2005), pp. 4925–4938.

[9] F. T. BÖLLE, A. E. G. MIKKELSEN, K. S. THYGESEN, T. VEGGE, AND I. E. CASTELLI, *Structural and chemical mechanisms governing stability of inorganic Janus nanotubes*, npj Computational Materials, 7 (2021).

[10] M. R. BONYADI AND Z. MICHALEWICZ, *Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review*, Evolutionary Computation, 25 (2017), pp. 1–54.

[11] K. BRAMAN, T. A. OLIVER, AND V. RAMAN, *Bayesian analysis of syngas chemistry models*, Combustion Theory and Modelling, 17 (2013), pp. 858–887.

[12] S. BROOKS, A. GELMAN, G. JONES, AND X.-L. MENG, eds., *Handbook of Markov Chain Monte Carlo*, Chapman and Hall/CRC, May 2011.

[13] S. P. BROOKS AND A. GELMAN, *General Methods for Monitoring Convergence of Iterative Simulations*, Journal of Computational and Graphical Statistics, 7 (1998), pp. 434–455.

[14] C. G. BROYDEN, *The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations*, IMA Journal of Applied Mathematics, 6 (1970), pp. 76–90.

[15] A. D. BULL, *Convergence rates of efficient global optimization algorithms*, 2011.

[16] L. CAI, H. PITSCH, S. Y. MOHAMED, V. RAMAN, J. BUGLER, H. CURRAN, AND S. M. SARATHY, *Optimized reaction mechanism rate rules for ignition of normal alkanes*, Combustion and Flame, 173 (2016), pp. 468–482.

[17] K. CHALONER AND I. VERDINELLI, *Bayesian Experimental Design: A Review*, Statistical Science, 10 (1995).

[18] E. CISNEROS-GARIBAY, C. PANTANO, AND J. B. FREUND, *Accounting for uncertainty in RCCE species selection*, Combustion and Flame, 208 (2019), pp. 219–234.

[19] M. A. CLYDE, J. O. BERGER, F. BULLARD, E. B. FORD, W. H. JEFFERYS, R. LUO, R. PAULO, AND T. LOREDO, *Current Challenges in Bayesian Model Choice*, in Statistical

Challenges in Modern Astronomy IV, G. J. Babu and E. D. Feigelson, eds., vol. 371 of Astronomical Society of the Pacific Conference Series, Nov. 2007, p. 224.

[20] V. L. COLVIN, M. C. SCHLAMP, AND A. P. ALIVISATOS, *Light-emitting diodes made from cadmium selenide nanocrystals and a semiconducting polymer*, Nature, 370 (1994), pp. 354–357.

[21] F. P. DA SILVA, J. L. FIORIO, AND L. M. ROSSI, *Tuning the Catalytic Activity and Selectivity of* Pd *Nanoparticles Using Ligand-Modified Supports and Surfaces*, ACS Omega, 2 (2017), pp. 6014–6022.

[22] P. DE AGUIAR, B. BOURGUIGNON, M. KHOTS, D. MASSART, AND R. PHAN-THAN-LUU, *D-optimal designs*, Chemometrics and Intelligent Laboratory Systems, 30 (1995), pp. 199–210.

[23] V. DUBOURG, *Adaptive surrogate models for reliability analysis and reliability-based design optimization*, PhD thesis, Université Blaise Pascal - Clermont-Ferrand II, 2011.

[24] D. ERIKSSON, K. DONG, E. LEE, D. BINDEL, AND A. G. WILSON, *Scaling Gaussian process regression with derivatives*, Advances in neural information processing systems, 31 (2018).

[25] M.-Y. FAN, Y.-L. ZHANG, Y.-C. LIN, J. LI, H. CHENG, N. AN, Y. SUN, Y. QIU, F. CAO, AND P. FU, *Roles of Sulfur Oxidation Pathways in the Variability in Stable Sulfur Isotopic Composition of Sulfate Aerosols at an Urban Site in Beijing, China*, Environmental Science & Technology Letters, 7 (2020), pp. 883–888.

[26] I. FAVIER, D. PLA, AND M. GÓMEZ, *Palladium Nanoparticles in Polyols: Synthesis, Catalytic Couplings, and Hydrogenations*, Chemical Reviews, 120 (2019), pp. 1146–1183.

[27] R. J. FIELD AND R. M. NOYES, *Oscillations in chemical systems. 18. Mechanisms of chemical oscillators: conceptual bases*, Accounts of Chemical Research, 10 (1977), pp. 214–221.

[28] E. E. FINNEY AND R. G. FINKE, *The Four-Step, Double-Autocatalytic Mechanism for Transition-Metal Nanocluster Nucleation, Growth, and Then Agglomeration: Metal, Ligand, Concentration, Temperature, and Solvent Dependency Studies*, Chemistry of Materials, 20 (2008), pp. 1956–1970.

[29] R. FLETCHER, *A new approach to variable metric algorithms*, The Computer Journal, 13 (1970), pp. 317–322.

[30] N. FRIEL AND J. WYSE, *Estimating the evidence - a review*, Statistica Neerlandica, 66 (2012), pp. 288–308.

[31] N. GALAGALI AND Y. M. MARZOUK, *Bayesian inference of chemical kinetic models from proposed reactions*, Chemical Engineering Science, 123 (2015), pp. 170–190.

[32] E. GALLICCHIO, M. ANDREC, A. K. FELTS, AND R. M. LEVY, *Temperature Weighted Histogram Analysis Method, Replica Exchange, and Transition Paths†*, The Journal of Physical Chemistry B, 109 (2005), pp. 6722–6731.

[33] D. C. GARY, M. W. TERBAN, S. J. L. BILLINGE, AND B. M. COSSAIRT, *Two-Step Nucleation and Growth of InP Quantum Dots via Magic-Sized Cluster Intermediates*, Chemistry of Materials, 27 (2015), pp. 1432–1441.

[34] D. J. GAVAGHAN, J. COOPER, A. C. DALY, C. GILL, K. GILLOW, M. ROBINSON, A. N. SIMONOV, J. ZHANG, AND A. M. BOND, *Use of Bayesian Inference for Parameter Recovery in DC and AC Voltammetry*, ChemElectroChem, 5 (2017), pp. 917–935.

[35] A. GELMAN, J. B. CARLIN, H. S. STERN, D. B. DUNSON, A. VEHTARI, AND D. B. RUBIN, *Bayesian Data Analysis*, Chapman & Hall/CRC Texts in Statistical Science, Chapman & Hall/CRC, Philadelphia, PA, 3 ed., Nov. 2013.

[36] A. GELMAN, W. R. GILKS, AND G. O. ROBERTS, *Weak convergence and optimal scaling of random walk Metropolis algorithms*, The Annals of Applied Probability, 7 (1997).

[37] A. GELMAN, G. O. ROBERTS, AND W. R. GILKS, *Efficient Metropolis Jumping Rules*, Clarendon Press, Oxford, England, Apr. 1996.

[38] D. GOLDFARB, *A family of variable-metric methods derived by variational means*, Mathematics of Computation, 24 (1970), pp. 23–26.

[39] J. GOODMAN AND J. WEARE, *Ensemble samplers with affine invariance*, Communications in Applied Mathematics and Computational Science, 5 (2010), pp. 65–80.

[40] G. GUENNEBAUD, B. JACOB, ET AL., *Eigen v3*. http://eigen.tuxfamily.org, 2010.

[41] N. GUNANTARA, *A review of multi-objective optimization: Methods and its applications*, Cogent Engineering, 5 (2018), p. 1502242.

[42] H. HAARIO, M. LAINE, A. MIRA, AND E. SAKSMAN, *DRAM: efficient adaptive MCMC*, Statistics and computing, 16 (2006), pp. 339–354.

[43] H. HAARIO, E. SAKSMAN, AND J. TAMMINEN, *An Adaptive Metropolis Algorithm*, Bernoulli, 7 (2001), p. 223.

[44] L. HAKIM, G. LACAZE, M. KHALIL, H. N. NAJM, AND J. C. OEFELEIN, *Modeling Auto-Ignition Transients in Reacting Diesel Jets*, Journal of Engineering for Gas Turbines and Power, 138 (2016).

[45] L. HAKIM, G. LACAZE, M. KHALIL, K. SARGSYAN, H. NAJM, AND J. OEFELEIN, *Probabilistic parameter estimation in a 2-step chemical kinetics model for n-dodecane jet autoignition*, Combustion Theory and Modelling, 22 (2018), pp. 446–466.

[46] D. HANDWERK, *Mechanism-Enabled Population Balances and the Effects of Anisotropies in the Complex Ginzburg-Landau Equation*, PhD thesis, Department of Mathematics, Colorado State University, Fort Collins, CO, 2019.

[47] D. R. HANDWERK, P. D. SHIPMAN, S. ÖZKAR, AND R. G. FINKE, *Dust Effects on Ir(0)n Nanoparticle Formation Nucleation and Growth Kinetics and Particle Size-Distributions:*

Analysis by and Insights from Mechanism-Enabled Population Balance Modeling, Langmuir, 36 (2020), pp. 1496–1506.

[48] D. R. HANDWERK, P. D. SHIPMAN, C. B. WHITEHEAD, S. ÖZKAR, AND R. G. FINKE, *Mechanism-Enabled Population Balance Modeling of Particle Formation en Route to Particle Average Size and Size Distribution Understanding and Control*, Journal of the American Chemical Society, 141 (2019), pp. 15827–15839.

[49] ——, *Particle Size Distributions via Mechanism-Enabled Population Balance Modeling*, The Journal of Physical Chemistry C, 124 (2020), pp. 4852–4880.

[50] W. K. HASTINGS, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.

[51] D. HIBBERT AND N. ARMSTRONG, *An introduction to Bayesian methods for analyzing chemistry data*, Chemometrics and Intelligent Laboratory Systems, 97 (2009), pp. 211–220.

[52] A. C. HINDMARSH, P. N. BROWN, K. E. GRANT, S. L. LEE, R. SERBAN, D. E. SHUMAKER, AND C. S. WOODWARD, *SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers*, ACM Transactions on Mathematical Software (TOMS), 31 (2005), pp. 363–396.

[53] M. D. HOFFMAN, A. GELMAN, ET AL., *The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.*, J. Mach. Learn. Res., 15 (2014), pp. 1593–1623.

[54] R. HOOKE AND T. A. JEEVES, `` *Direct Search'' Solution of Numerical and Statistical Problems*, Journal of the ACM, 8 (1961), pp. 212–229.

[55] B. J. HORNSTEIN AND R. G. FINKE, *Transition-Metal Nanocluster Kinetic and Mechanistic Studies Emphasizing Nanocluster Agglomeration: Demonstration of a Kinetic Method That Allows Monitoring of All Three Phases of Nanocluster Formation and Aging*, Chemistry of Materials, 16 (2003), pp. 139–150.

[56] H. JEFFREYS, *The Theory of Probability*, Oxford Classic Texts in the Physical Sciences, OUP Oxford, 1998.

[57] R. H. JOHNSTONE, E. T. CHANG, R. BARDENET, T. P. DE BOER, D. J. GAVAGHAN, P. PATHMANATHAN, R. H. CLAYTON, AND G. R. MIRAMS, *Uncertainty and variability in models of the cardiac action potential: Can we build trustworthy models?*, Journal of Molecular and Cellular Cardiology, 96 (2016), pp. 49–62.

[58] D. R. JONES, M. SCHONLAU, AND W. J. WELCH, *Efficient Global Optimization of Expensive Black-Box Functions*, Journal of Global Optimization, 13 (1998), pp. 455–492.

[59] H. KANEKO AND K. FUNATSU, *Adaptive soft sensor based on online support vector regression and Bayesian ensemble learning for various states in chemical plants*, Chemometrics and Intelligent Laboratory Systems, 137 (2014), pp. 57–66.

[60] R. E. KASS AND A. E. RAFTERY, *Bayes Factors*, Journal of the American Statistical Association, 90 (1995), pp. 773–795.

[61] J. KENNEDY AND R. EBERHART, *Particle swarm optimization*, in Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, 1995.

[62] P. D. KENT, J. E. MONDLOCH, AND R. G. FINKE, *A Four-Step Mechanism for the Formation of Supported-Nanoparticle Heterogenous Catalysts in Contact with Solution: The Conversion of Ir (1,5-COD) Cl/*γ-$\mathrm{Al_2O_3}$ *to* $\mathrm{Ir(0)_{\sim 170}/}\gamma$-$\mathrm{Al_2O_3}$, Journal of the American Chemical Society, 136 (2014), pp. 1930–1941.

[63] S. KLEINEGESSE, C. DROVANDI, AND M. U. GUTMANN, *Sequential Bayesian Experimental Design for Implicit Models via Mutual Information*, 2020.

[64] S. KLEINEGESSE AND M. U. GUTMANN, *Efficient Bayesian Experimental Design for Implicit Models*, in Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, K. Chaudhuri and M. Sugiyama, eds., vol. 89 of Proceedings of Machine Learning Research, PMLR, 16–18 Apr 2019, pp. 476–485.

[65] T. G. KOLDA, R. M. LEWIS, AND V. TORCZON, *Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods*, SIAM Review, 45 (2003), pp. 385–482.

[66] D. G. KRIGE, *A statistical approach to some basic mine valuation problems on the Witwatersrand*, Journal of the Southern African Institute of Mining and Metallurgy, 52 (1951), pp. 119–139.

[67] S. KULLBACK AND R. A. LEIBLER, *On Information and Sufficiency*, The Annals of Mathematical Statistics, 22 (1951), pp. 79–86.

[68] H. J. KUSHNER, *A versatile stochastic model of a function of unknown and time varying form*, Journal of Mathematical Analysis and Applications, 5 (1962), pp. 150–167.

[69] H. O. LLOYD-LANEY, N. D. J. YATES, M. J. ROBINSON, A. R. HEWSON, J. D. FIRTH, D. M. ELTON, J. ZHANG, A. M. BOND, A. PARKIN, AND D. J. GAVAGHAN, *Using Purely Sinusoidal Voltammetry for Rapid Inference of Surface-Confined Electrochemical Reaction Parameters*, Analytical Chemistry, 93 (2021), pp. 2062–2071.

[70] D. K. LONG, W. BANGERTH, D. R. HANDWERK, C. B. WHITEHEAD, P. D. SHIPMAN, AND R. G. FINKE, *Estimating reaction parameters in mechanism-enabled population balance models of nanoparticle size distributions: A Bayesian inverse problem approach*, Journal of Computational Chemistry, 43 (2022), pp. 43–56.

[71] A. MAKAROVA, H. SHEN, V. PERRONE, A. KLEIN, J. B. FADDOUL, A. KRAUSE, M. SEEGER, AND C. ARCHAMBEAU, *Automatic Termination for Hyperparameter Optimization*, 2021.

[72] S. MARELLI AND B. SUDRET, *UQLab: A Framework for Uncertainty Quantification in Matlab*, in Vulnerability, Uncertainty, and Risk, American Society of Civil Engineers, June 2014.

[73] B. MATÉRN, *Spatial Variation*, Springer New York, 1986.

[74] MATHWORKS, *Statistics and Machine Learning Toolbox: bayesopt [R2022b]*. https://www.mathworks.com/help/stats/bayesopt.html, 2016. Accessed: 2023-02-17.

[75] ——, *Statistics and Machine Learning Toolbox: Bayesian Optimization Algorithm [R2022b]*. https://www.mathworks.com/help/stats/bayesian-optimization-algorithm.html, 2023. Accessed: 2023-02-17.

[76] R. M. NEAL, *Probabilistic inference using Markov chain Monte Carlo methods*, tech. rep., Department of Computer Science, University of Toronto Toronto, Ontario, Canada, 1993.

[77] J. A. NELDER AND R. MEAD, *A Simplex Method for Function Minimization*, The Computer Journal, 7 (1965), pp. 308–313.

[78] M. A. NEWTON AND A. E. RAFTERY, *Approximate Bayesian Inference with the Weighted Likelihood Bootstrap*, Journal of the Royal Statistical Society, 56 (1994), pp. 3–48.

[79] V. NGUYEN, S. GUPTA, S. RANA, C. LI, AND S. VENKATESH, *Regret for Expected Improvement over the Best-Observed Value and Stopping Condition*, in Proceedings of the Ninth Asian Conference on Machine Learning, M.-L. Zhang and Y.-K. Noh, eds., vol. 77 of Proceedings of Machine Learning Research, Yonsei University, Seoul, Republic of Korea, 15–17 Nov 2017, PMLR, pp. 279–294.

[80] S. OLADYSHKIN AND W. NOWAK, *Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion*, Reliability Engineering & System Safety, 106 (2012), pp. 179–190.

[81] S. ÖZKAR AND R. G. FINKE, *Nanoparticle Nucleation Is Termolecular in Metal and Involves Hydrogen: Evidence for a Kinetically Effective Nucleus of Three* $\{\mathrm{Ir_3\,H_{2\mathit{x}}}\cdot\mathrm{P_2\,W_{15}\,Nb_3\,O_{62}}\}^{6-}$ *in* $\mathrm{Ir(0)}_n$ *Nanoparticle Formation From* $[(1{,}5\text{-COD})\mathrm{Ir^I}\cdot\mathrm{P_2\,W_{15}\,Nb_3\,O_{62}}]^{8-}$ *plus dihydrogen*, Journal of the American Chemical Society, 139 (2017), pp. 5444–5457.

[82] E. PARZEN, *On Estimation of a Probability Density Function and Mode*, The Annals of Mathematical Statistics, 33 (1962), pp. 1065–1076.

[83] A. G. PATTANTYUS-ABRAHAM, I. J. KRAMER, A. R. BARKHOUSE, X. WANG, G. KONSTANTATOS, R. DEBNATH, L. LEVINA, I. RAABE, M. K. NAZEERUDDIN, M. GRÄTZEL, AND E. H. SARGENT, *Depleted-Heterojunction Colloidal Quantum Dot Solar Cells*, ACS Nano, 4 (2010), pp. 3374–3380.

[84] J. PRAGER, H. N. NAJM, K. SARGSYAN, C. SAFTA, AND W. J. PITZ, *Uncertainty quantification of reaction mechanisms accounting for correlations introduced by rate rules and fitted Arrhenius parameters*, Combustion and Flame, 160 (2013), pp. 1583–1593.

[85] L. PROTESESCU, S. YAKUNIN, M. I. BODNARCHUK, F. KRIEG, R. CAPUTO, C. H. HENDON, R. X. YANG, A. WALSH, AND M. V. KOVALENKO, *Nanocrystals of Cesium Lead Halide Perovskites (CsPbX3, X = Cl, Br, and I): Novel Optoelectronic Materials Showing Bright Emission with Wide Color Gamut*, Nano Letters, 15 (2015), pp. 3692–3696.

[86] A. PULIYANDA, K. SIVARAMAKRISHNAN, Z. LI, A. DE KLERK, AND V. PRASAD, *Data fusion by joint non-negative matrix factorization for hypothesizing pseudo-chemistry using Bayesian networks*, Reaction Chemistry & Engineering, 5 (2020), pp. 1719–1737.

[87] S. RANFTL AND W. VON DER LINDEN, *Bayesian Surrogate Analysis and Uncertainty Propagation*, in The 40th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, MDPI, Nov. 2021.

[88] C. E. RASMUSSEN AND C. K. I. WILLIAMS, *Gaussian processes for machine learning*, Adaptive Computation and Machine Learning series, MIT Press, London, England, Nov. 2005.

[89] C. P. ROBERT, *The expected demise of the Bayes factor*, Journal of Mathematical Psychology, 72 (2016), pp. 33–37.

[90] M. ROSENBLATT, *Remarks on Some Nonparametric Estimates of a Density Function*, The Annals of Mathematical Statistics, 27 (1956), pp. 832–837.

[91] E. G. RYAN, C. C. DROVANDI, J. M. MCGREE, AND A. N. PETTITT, *A Review of Modern Computational Algorithms for Bayesian Optimal Design*, International Statistical Review, 84 (2015), pp. 128–154.

[92] Y. SAAD, *ILUT: A dual threshold incomplete LU factorization*, Numerical Linear Algebra with Applications, 1 (1994), pp. 387–402.

[93] T. J. SANTNER, B. J. WILLIAMS, AND W. I. NOTZ, *The Design and Analysis of Computer Experiments*, Springer New York, 2003.

[94] K. SARGSYAN, H. N. NAJM, AND R. GHANEM, *On the Statistical Calibration of Physical Models*, International Journal of Chemical Kinetics, 47 (2015), pp. 246–276.

[95] R. D. SCHALLER AND V. I. KLIMOV, *High Efficiency Carrier Multiplication in PbSe Nanocrystals: Implications for Solar Energy Conversion*, Physical Review Letters, 92 (2004).

[96] A. SCHMIDT AND V. SMIRNOV, *Concept of "magic" number clusters as a new approach to the interpretation of unusual kinetics of the Heck reaction with aryl bromides*, Topics in Catalysis, 32 (2005), pp. 71–75.

[97] L. F. SHAMPINE AND M. W. REICHELT, *The MATLAB ODE suite*, SIAM Journal on Scientific Computing, 18 (1997), pp. 1–22.

[98] D. F. SHANNO, *Conditioning of quasi-newton methods for function minimization*, Mathematics of Computation, 24 (1970), pp. 647–656.

[99] C. E. SHANNON, *A Mathematical Theory of Communication*, Bell System Technical Journal, 27 (1948), pp. 379–423.

[100] W. SHAO AND X. TIAN, *Adaptive soft sensor for quality prediction of chemical processes based on selective ensemble of local partial least squares models*, Chemical Engineering Research and Design, 95 (2015), pp. 113–132.

[101] Y. SHIRASAKI, G. J. SUPRAN, M. G. BAWENDI, AND V. BULOVIĆ, *Emergence of colloidal quantum-dot light-emitting technologies*, Nature Photonics, 7 (2012), pp. 13–23.

[102] K. SMITH, *On the Standard Deviations of Adjusted and Interpolated Values of an Observed Polynomial Function and its Constants and the Guidance they give Towards a Proper Choice of the Distribution of Observations*, Biometrika, 12 (1918), p. 1.

[103] R. C. SMITH, *Uncertainty Quantification: Theory, Implementation, and Applications*, Society for Industrial and Applied Mathematics, 2014.

[104] I. SPANOPOULOS, I. HADAR, W. KE, P. GUO, E. M. MOZUR, E. MORGAN, S. WANG, D. ZHENG, S. PADGAONKAR, G. N. M. REDDY, E. A. WEISS, M. C. HERSAM, R. SESHADRI, R. D. SCHALLER, AND M. G. KANATZIDIS, *Tunable Broad Light Emission from 3D "Hollow" Bromide Perovskites through Defect Engineering*, Journal of the American Chemical Society, 143 (2021), pp. 7069–7080.

[105] M. L. STEIN, *Interpolation of Spatial Data*, Springer New York, 1999.

[106] M. STONE, *Cross-Validatory Choice and Assessment of Statistical Predictions*, Journal of the Royal Statistical Society: Series B (Methodological), 36 (1974), pp. 111–133.

[107] J. W. STOUWDAM AND R. A. J. JANSSEN, *Red, green, and blue quantum dot LEDs with solution processable ZnO nanocrystal electron injection layers*, Journal of Materials Chemistry, 18 (2008), p. 1889.

[108] N. SUN, R. J. CARROLL, AND H. ZHAO, *Bayesian error analysis model for reconstructing transcriptional regulatory networks*, Proceedings of the National Academy of Sciences, 103 (2006), pp. 7988–7993.

[109] O. THOMAS, R. DUTTA, J. CORANDER, S. KASKI, AND M. U. GUTMANN, *Likelihood-free inference by ratio estimation*, 2016.

[110] H. A. VAN DER VORST, *Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems*, SIAM Journal on Scientific and Statistical Computing, 13 (1992), pp. 631–644.

[111] M. VERT, Y. DOI, K.-H. HELLWICH, M. HESS, P. HODGE, P. KUBISA, M. RINAUDO, AND F. SCHUÉ, *Terminology for biorelated polymers and applications (IUPAC recommendations 2012)*, Pure and Applied Chemistry, 84 (2012), pp. 377–410.

[112] M. D. VOSE, *Modeling Simple Genetic Algorithms*, in Foundations of Genetic Algorithms, Elsevier, 1993, pp. 63–73.

[113] J. A. VRUGT, C. TER BRAAK, C. DIKS, B. A. ROBINSON, J. M. HYMAN, AND D. HIGDON, *Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling*, International Journal of Nonlinear Sciences and Numerical Simulation, 10 (2009).

[114] P.-R. WAGNER, J. NAGEL, S. MARELLI, AND B. SUDRET, *UQLab user manual – Bayesian inversion for model calibration and validation*, tech. rep., Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich,Switzerland, 2022. Report UQLab-V2.0-113.

[115] M. A. WATZKY AND R. G. FINKE, *Nanocluster Size-Control and "Magic Number" Investigations. Experimental Tests of the "Living-Metal Polymer" Concept and of Mechanism-Based Size-Control Predictions Leading to the Syntheses of Iridium(0) Nanoclusters Centering about Four Sequential Magic Numbers*, Chemistry of Materials, 9 (1997), pp. 3083–3095.

[116] M. A. WATZKY AND R. G. FINKE, *Transition metal nanocluster formation kinetic and mechanistic studies. A new mechanism when hydrogen is the reductant: slow, continuous*

nucleation and fast autocatalytic surface growth, Journal of the American Chemical Society, 119 (1997), pp. 10382–10400.

[117] M. A. WATZKY, E. E. FINNEY, AND R. G. FINKE, *Transition-Metal Nanocluster Size vs Formation Time and the Catalytically Effective Nucleus Number: A Mechanism-Based Treatment*, Journal of the American Chemical Society, 130 (2008), pp. 11959–11969.

[118] C. B. WHITEHEAD, D. R. HANDWERK, P. D. SHIPMAN, Y. LI, A. I. FRENKEL, B. INGHAM, N. M. KIRBY, AND R. G. FINKE, *Nanoparticle Formation Kinetics, Mechanisms, and Accurate Rate Constants: Examination of a Second-Generation Ir(0)n Particle Formation System by Five Monitoring Methods Plus Initial Mechanism-Enabled Population Balance Modeling*, The Journal of Physical Chemistry C, 125 (2021), pp. 13449–13476.

[119] J. WU, M. POLOCZEK, A. G. WILSON, AND P. FRAZIER, *Bayesian optimization with gradients*, Advances in neural information processing systems, 30 (2017).

[120] D. XIU AND G. E. KARNIADAKIS, *The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations*, SIAM Journal on Scientific Computing, 24 (2002), pp. 619–644.

[121] J. ZÁDOR, I. G. ZSÉLY, T. TURÁNYI, M. RATTO, S. TARANTOLA, AND A. SALTELLI, *Local and Global Uncertainty Analyses of a Methane Flame Model*, The Journal of Physical Chemistry A, 109 (2005), pp. 9795–9807.